DIGITAL GIGAswitch/Router

User Reference Manual

Part Number: 9032684-03

December 1999

This manual describes how to use the DIGITAL GIGAswitch/Router (GSR).

Revision/Update Information: This is a revised document.

Changes

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.

Disclaimer

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.

Copyright

Printed in the United States of America

Trademarks

Apple, AppleTalk, and Macintosh are registered trademarks of Apple Computer, Inc.

Cabletron Systems is registered trademark and Cabletron, clearVISN, and GIGAswitch are trademarks of Cabletron Systems, Inc.

EtherChannel is a registered trademark of Cicsco Systems, Inc.

DIGITAL Equipment Corporation, DEC, and the DIGITAL Equipment Corporation logo are registered trademarks and DECnet is a trademark of DIGITAL Equipment Corporation Equipment Corporation.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Java is a trademark of Sun Microsytems, Inc.

All other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

FCC Notice — Class A Computing Device

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference. Any modifications to this device - unless expressly approved by the manufacturer - can void the user's authority to operate this equipment under part 15 of the FCC rules.

Industry Canada Notice

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

NOTICE: The Industry Canada label identifies certified equipment. This certification means that the equipment meets telecommunications network protective, operational and safety requirements as prescribed in the appropriate Terminal Equipment Technical Requirements document(s). The department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be coordinated by a representative designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas. **Caution:** Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

NOTICE: The Ringer Equivalence Number (REN) assigned to each terminal device provides an indication of the maximum number of terminals allowed to be connected to a telephone interface. The termination on an interface may consist of any combination of devices subject only to the requirement that the sum of the ringer equivalence Numbers of all the devices does not exceed 5.

VCCI Notice — Class A Computing Device

This equipment is a Class A product (information equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council for Interference by Data Processing Equipment and Electronic Office Machines aimed at preventing radio interference in commercial and/or industrial areas. Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers. Read the instructions for correct handling.

この装置は，情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Taiwanese Notice — Class A Computing Device

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

CE Notice — Class A Computing Device

Warning!
This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may be required to take adequate measures.

Achtung!
Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Avertissement!
Cet appareil est un appareil de Classe A. Dans un environnement résidentiel cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l'utilisateur de prendre les mesures appropriées.

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 program (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 in writing 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.

SAFETY INFORMATION

CLASS 1 LASER TRANSCEIVERS

The DGSRF-AA 100Base-FX Module, DGSRS-AA 1000Base-LX Module, and DGSRL-AA 1000Base-LX Module use Class 1 Laser transceivers. Read the following safety information before installing or operating these modules.

The Class 1 laser transceivers use an optical feedback loop to maintain Class 1 operation limits. This control loop eliminates the need for maintenance checks or adjustments. The output is factory set, and does not allow any user adjustment. Class 1 Laser transceivers comply with the following safety standards:

- 21 CFR 1040.10 and 1040.11 U.S. Department of Health and Human Services (FDA).
- IEC Publication 825 (International Electrotechnical Commission).
- CENELEC EN 60825 (European Committee for Electrotechnical Standardization).

When operating within their performance limitations, laser transceiver output meets the Class 1 accessible emission limit of all three standards. Class 1 levels of laser radiation are not considered hazardous.

Laser Radiation and Connectors

When the connector is in place, all laser radiation remains within the fiber. The maximum amount of radiant power exiting the fiber (under normal conditions) is -12.6 dBm or 55×10^{-6} watts.

Removing the optical connector from the transceiver allows laser radiation to emit directly from the optical port. The maximum radiance from the optical port (under worst case conditions) is $0.8\ W\ cm^{-2}$ or $8 \times 10^{3}\ W\ m^{2}\ sr\text{-}1$.

Do not use optical instruments to view the laser output. The use of optical instruments to view laser output increases eye hazard. When viewing the output optical port, power must be removed from the network adapter.

DECLARATION OF CONFORMITY

Application of Council Directive(s):	**89/336/EEC** **73/23/EEC**
Manufacturer's Name:	**Cabletron Systems, Inc.**
Manufacturer's Address:	**35 Industrial Way** PO Box 5005 Rochester, NH 03867
European Representative Name:	**Mr. J. Solari**
European Representative Address:	**Cabletron Systems Limited** **Nexus House, Newbury** **Business Park** **London Road, Newbury** **Berkshire RG13 2PZ, England**
Conformance to Directive(s)/Product Standards:	**EC Directive 89/336/EEC** **EC Directive 73/23/EEC** **EN 55022** **EN 50082-1** **EN 60950**
Equipment Type/Environment:	**Networking Equipment, for use in a Commercial or Light Industrial Environment.**

We the undersigned, hereby declare, under our sole responsibility, that the equipment packaged with this notice conforms to the above directives.

Manufacturer	Legal Representative in Europe
Mr. Ronald Fotino	Mr. J. Solari
Full Name	Full Name
Principal Compliance Engineer	Managing Director - E.M.E.A.
Title	Title
Rochester, NH, USA	Newbury, Berkshire, England
Location	Location

Contents

Preface

About This Manual

This manual provides detailed information and procedures for configuring the DIGITAL® GIGAswitch™/Router software. If you have not yet installed the GSR, 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.

Who Should Read This Manual?

Read this manual if you are a network administrator responsible for configuring and monitoring the GSR.

How to Use This Manual

If You Want To	See
Read overview information	Chapter 1, "DIGITAL GIGAswitch/Router Product Overview"
Hot swap line cards and Control Modules	Chapter 2, "Hot Swapping Line Cards and Control Modules"
Configure bridging	Chapter 3, "Bridging Configuration Guide"
Configure SmartTRUNKs	Chapter 4, "SmartTRUNK Configuration Guide"
Configure Dynamic Host Configuration Protocol server	Chapter 5, "DHCP Configuration Guide"
Configure IP interfaces and global routing parameters	Chapter 6, "IP Routing Configuration Guide"
Configure VRRP	Chapter 7, "VRRP Configuration Guide"
Configure RIP routing	Chapter 8, "RIP Configuration Guide"

If You Want To	See
Configure OSPF routing	Chapter 9, ”OSPF Configuration Guide”
Configure BGP routing	Chapter 10, ”BGP Configuration Guide”
Configure routing policies	Chapter 11, ”Routing Policy Configuration Guide”
Configure IP multicast routing	Chapter 12, ”Multicast Routing Configuration Guide”
Configure IP policy-based forwarding	Chapter 13, ”IP Policy-Based Forwarding Configuration Guide”
Configure Network Address Translation	Chapter 14, ”Network Address Translation Configuration Guide”
Configure web hosting	Chapter 15, ”Web Hosting Configuration Guide”
Configure IPX routing	Chapter 16, ”IPX Routing Configuration Guide”
Configure Access Control Lists	Chapter 17, ”Access Control List Configuration Guide”
Configure security	Chapter 18, ”Security Configuration Guide”
Configure QoS (Quality of Service) parameters	Chapter 19, ”QoS Configuration Guide”
Monitor performance	Chapter 20, ”Performance Monitoring Guide”
Configure RMON	Chapter 21, ”RMON Configuration Guide”
Configure WAN	Chapter 22, ”WAN Configuration Guide”

Related Documentation

The DIGITAL GIGAswitch/Router 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 GSR	*DIGITAL GIGAswitch/Router Getting Started Guide*
Managing the GSR using DIGITAL's 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 Reference Manual*

Correspondence

Documentation Comments

If you have comments or suggestions about this manual, send them to the DIGITAL Network Products Organization.

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

Online Services

To locate product-specific information, refer to the DIGITAL Network Products Home Page on the World Wide Web located at the following addresses:

Americas: http://www.networks.digital.com

Europe: http://www.networks.europe.digital.com

Asia Pacific: http://www.networks.digital.com.au

Getting Help

To expedite your inquiry when you contact your DIGITAL representative, please provide the following information:

- Your Name
- Your Company Name
- Address
- Email Address
- Phone Number
- FAX Number
- Detailed description of the issue (including history, what you've tried, and conditions under which you see this occur)
- Hardware module number, software version, and switch configuration (that is, what part types are in what slots)

Chapter 1

DIGITAL GIGAswitch/Router Product Overview

The DIGITAL GIGAswitch/Router 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 GSR:

Table 1. GSR **Hardware and software specifications**

Feature	Specification
Throughput	GSR-8: • 16-Gbps non-blocking switching fabric • Up to 15 million packets-per-second routing throughput
	GSR-16: • 32-Gbps non-blocking switching fabric • Up to 30 million packets-per-second routing throughput

Table 1. GSR **Hardware and software specifications (Continued)**

Feature	Specification
Capacity	• 4,096 Virtual LANs (VLANs) • 3 MB input/output buffering per Gigabit port • 1 MB input/output buffering per 10/100 port
	GSR-8: • Up to 250,000 routes • Up to 2,000,000 Layer-4 application flows • Up to 400,000 Layer-2 MAC addresses • 20,000 Layer-2 security and access-control filters
	GSR-16: • Up to 250,000 routes • Up to 4,000,000 Layer-4 application flows • Up to 800,000 Layer-2 MAC addresses • 20,000 Layer-2 security and access-control filters
Routing protocols	• IP: RIP v1/v2, OSPF, BGP 2, 3, 4 • 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)
Quality of Service (QoS)	• Layer-2 prioritization (802.1p) • Layer-3 source-destination flows • Layer-4 source-destination flows • Layer-4 application flows
RMON	• RMON v1/v2 for each port

Table 1. GSR **Hardware and software specifications (Continued)**

Feature	Specification
Management	• SNMP • clearVISN™ 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)
Load balancing/ sharing	• Cabletron Systems® SmartTRUNK support
Redundancy	• Redundant and hot-swappable power supplies • Virtual Router Redundancy Protocol (VRRP)

Supported Media (Encapsulation Type)

The GSR 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 GSR supports many routing protocols based on open standards. The GSR 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

  Chapter 6, "IP Routing Configuration Guide," describes these protocols in detail.
- Exterior gateway protocol:
  - Border Gateway Protocol (BGP) Version 2,3,4

  Chapter 10, "BGP Configuration Guide," describes this protocol in detail.

- Novell IPX routing protocols:
  - Routing Information Protocol (RIP)
  - Service Advertising Protocol (SAP)

  Chapter 16, "IPX Routing Configuration Guide," describes these protocols in detail.

Configuring the DIGITAL GIGAswitch/Router

The GSR provides a command line interface (CLI) that allows you to configure and manage the GSR. The CLI has several command modes, each of which provides a group of related commands that you can use to configure the GSR 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 GSR 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.

GSR commands can be entered at a terminal connected to the access server or router using the command line interface (CLI). The GSR can also be configured using the DIGITAL clearVISN CoreWatch Java-based management application. Using DIGITAL clearVISN 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.

Table 2. Common CLI key commands

Key Sequence	Command
Ctrl+A	Move cursor to beginning of line
Ctrl+B	Move cursor back one character
Ctrl+D	Delete character

Table 2. Common CLI key commands (Continued)

Key Sequence	Command
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 GSR 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 the > 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 the # 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 `(config)`.
- **Boot** – This mode appears when the GSR the external flash card or the system image is not found during bootup. You should enter the **reboot** command to reset the GSR. If the GSR still fails to bootup, please call DIGITAL Technical Support.

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, enter 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 GSR, 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 GSR name followed by the angle bracket (>):

```
gs/r>
```

The default name is GSR 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
 help                 - Describe online help facility
 igmp                 - Show IGMP related parameters
 ip-redundancy        - Show IP Redundancy information (VRRP)
 ipx                  - Show IPX related parameters
 l2-tables            - Show L2 Tables information
 logout               - Log off the system
 multicast            - Configure Multicast related parameters
 ping                 - Ping utility
 pvst                 - Show Per Vlan Spanning Tree Protocol (PVST)
                          parameters
 sfs                  - Show SecureFast Switching (SFS) parameters
 statistics           - Show or clear GSR statistics
 stp               - Show STP status
 telnet            - Telnet utility
 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 Enable commands.	**?**

The Enable mode command prompt consists of the GSR name 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
 bgp                - Show Border Gateway Protocol (BGP) parameters
 cli                - Modify the command line interface behavior
 configure          - Enter Configuration Mode
 copy               - Copy configuration database
 dhcp               - Configure DHCP server
 dvmrp              - Show DVMRP related parameters
 enable             - Enable privileged user mode
 exit               - Exit current mode
 file               - File manipulation commands
 filters            - Show L2 security filters
 frame-relay        - Display Frame Relay statistics
 help               - Describe online help facility
 http               - Show http parameters
 igmp               - Show IGMP related parameters
 interface          - Show interface related parameters
 ip                 - Show IP related parameters
 ip-policy          - Show IP policy information
 ip-redundancy      - Show IP Redundancy information (VRRP)
 ip-router          - Show unicast IP Routing related parameters
 ipx                - Show IPX related parameters
 l2-tables          - Show L2 Tables information
 lfap                  - Show LFAP parameters
 load-balance          - Show Load Balancing related parameters and
                         hosts
 logout                - Log off the system
 mtrace                - Multicast Traceroute utility
 multicast             - Configure Multicast related parameters
 nat                   - Show Network Address Translation related
                         parameters
 ntp                   - Network Time Protocol (NTP)
 ospf                  - Show/Monitor Open Shortest Path First Protocol
                         (OSPF).
 ping                  - Ping utility
 port                  - Show or change Port parameters
 ppp                   - Display Point to Point Protocol (PPP)
                         statistics
 pvst                  - Show Per Vlan Spanning Tree Protocol (PVST)
                         parameters
 qos                   - Show Quality of Service parameters
 radius                - Show RADIUS related parameters
 rate-limit            - Show rate-limit policy information
 rdisc                 - Show Router Discovery Protocol (RIP) parameters
 reboot                - Reboot the system
 rip                   - Show/Query Routing Information Protocol(RIP)
                         tables
 rmon                  - Show RMON related parameters
 sfs                   - Show SecureFast Switching (SFS) parameters
```

```
 smarttrunk            - Show SmartTRUNK information
 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
 tacacs-plus           - Show TACACS+ related parameters
 telnet                - Telnet utility
 traceroute            - Traceroute utility
 vlan                  - Show VLAN-related parameters
 web-cache              - Configure web caching 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 GSR. 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 GSR 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
 acl-policy            - Configure ACL policy
 aging                 - Configure L2 and L3 Aging
 arp                   - Configure ARP entries
 bgp                   - Configure Border Gateway Protocol (BGP)
 cli                   - Modify the command line interface behavior
 dhcp                  - Configure DHCP server
 dvmrp                 - Configure DVMRP related parameters
 exit                  - Exit current mode
```

```
 filters               - Configure L2 security filters
 frame-relay           - Configure wan interface parameters
 help                  - Describe online help facility
 igmp                  - Configure IGMP related parameters
 interface             - Configure interface related parameters
 ip                    - Configure IP related parameters
 ip-policy             - Configure IP policy for packet forwarding
 ip-redundancy         - Configure IP redundancy protocols
 ip-router             - Configure Unicast Routing Protocol related
                          parameters
 ipx                   - Configure IPX related parameters
 lfap                  - Configure Lightweight Flow Accounting Protocol
                           client
 load-balance          - Configure Load Balancing related parameters
 nat                   - configure network address translation
                           parameters
 ntp                   - Configure Network Time Protocol (NTP)
                           parameters
 ospf                  - Configure Open Shortest Path Protocol (OSPF)
 port                  - Configure Port parameters
 ppp                   - Configure wan interface parameters
 pvst                  - Configure Per Vlan Spanning Tree Protocol
                           (PVST)
 qos                   - Configure Quality of Service parameters
 radius                - Configure RADIUS related parameters
 rate-limit            - Configure rate limits for flows
 rdisc                 - Configure Router Discovery Protocol
 rip                   - Configure Routing Information Protocol (RIP)
 rmon                  - Configure RMON related parameters
 sfs                   - Configure SecureFast Switching (SFS) parameters
 smarttrunk            - Configure SmartTRUNK
 snmp                  - Configure SNMP related parameters.
 stp                   - Configure STP parameters
 system                - Configure system-wide parameters
 tacacs                - Configure TACACS related parameters
 tacacs-plus           - Configure TACACS+ related parameters
 vlan                  - Configure VLAN-related parameters
 web-cache             - Configure web caching parameters

Special configuration mode commands:
 clear                 - Show configuration commands
 diff                  - Compare active configuration against another
                           configuration
 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 GSR 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 GSR at the boot PROM to restart the system. If the system fails to reboot successfully, please call DIGITAL Equipment Corporation Technical Support to resolve the problem.

To reboot the GSR 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 GSR contains an internal flash on the Control Module and an external PC flash. The internal flash contains the GSR boot image and user defined configuration files. An external PC flash contains the system image executed by the Control Module. When a GSR 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 GSR Control Module. Multiple system images can be stored on the external PC flash.

Configuration Files

The GSR 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 GSR without saving the active configuration changes to the Startup configuration file, the changes are lost.

- **Startup** – The configuration file that the GSR 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 GSR 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 GSR sorts out the dependencies and executes the command in the proper sequence.

Loading System Image Software

By default, the GSR 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 **system show version** command:

   Here is an example:

```
gs/r# system show version
Software Information
  Software Version   : 2.1
  Copyright          : Copyright (c) 1996-1998 Cabletron Systems, Inc.
  Image Information  : Version 2.1.0.0 built on Wed Jan 20 19:28:49 1999
  Image Boot Location: file:/pc-flash/boot/img8/
```

Note: In this example, the location "pc-flash" indicates that the GSR 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 GSR can access. (Use the **ping** command to verify that the GSR can reach the TFTP server.)
3. Use the **system image add** command to copy the software upgrade onto the PCMCIA flash card in the Control Module.

Here is an example:

```
gs/r# system image add 10.50.11.12 img2100
Downloading image 'img2100' from host '10.50.11.12'
 to local image img2100 (takes about 3 minutes)
kernel: 100%
Image checksum validated.
Image added.
```

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

   Here is an example:

```
gs/r# system image list
Images currently available:
img2100
```

5. Use the **system image choose** command to select the image file the GSR will use the next time you reboot the switch.

   Here is an example:

```
gs/r# system image choose img2100
Making image img2100 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 GSR 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 **system show version** command:

   Here is an example:

```
gs/r# system show version
Software Information
Software Version   : 2.1
Copyright          : Copyright (c) 1996-1999 Cabletron Systems, Inc.
Image Information  : Version 2.1.0.0, built on Wed Jan 2022:49:07 1999
Image Boot Location: file:/pc-flash/boot/img2100/
Boot Prom Version  : prom-1.0
```

   In this example, the location "pc-flash" indicates that the GSR 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 GSR can access. (Use the **ping** command to verify that the GSR can reach the TFTP server.)

3. Use the **system promimage upgrade** command to copy the boot PROM upgrade onto the internal memory in the Control Module.

   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 **system show version** command to verify that the new boot PROM software is on the internal memory of the Control Module:

Activating the 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
```

4. The CLI displays the following message:

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

5. Enter **yes** or **y** 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 the 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 GSR. However, if you power down or reboot the GSR, the new changes are lost. Use the following procedure to save the changes into the Startup configuration file so that the GSR reinstates the changes when you reboot the software.

1. Ensure that you are in the Enable mode by entering the **enable** command.
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** or **y** 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 **save startup** command:

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

Displaying Configuration Changes

While in Configure mode, you can display the configuration of the running system as well as non-activated changes that are in the Scratchpad by entering the following command:

Display running system configuration and non-activated changes in scratchpad.	**show**

While in Enable mode, you can display the active configuration of the system by entering the following command:

Display active configuration of the system.	**system show active-config**

The **show** and **system show active-config** commands normally display configuration commands in the order that they are executed. To display the configuration commands in a different order, enter the following command in Configure mode:

Display configuration commands in alphabetical order.	**system set show-config alphabetical**

Whenever you have activated commands in the scratchpad, you can compare the activated changes with a previously-saved configuration file. To compare the activated commands with the Startup (or another) configuration file, enter the following command in Configure mode:

Compare activated commands with Startup configuration file.	**diff <***filename***>|startup**

Managing the GSR

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

- Setting the GSR name
- Setting the GSR date and time
- Configuring NTP
- Configuring the CLI
- Configuring SNMP services
- Configuring DNS
- Connecting between the GSR and other systems

Setting the GSR Name

The GSR name is set to **gs/r** by default. You may customize the name for the GSR by entering the following command in Configure mode:

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

Setting GSR Date and Time

The GSR system time can keep track of time as entered by the user or via NTP. To configure the GSR date and time manually, 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 NTP

You can use the **ntp set server** command to instruct the GSR's NTP client to periodically synchronize its clock. By default, the GSR specifies an NTPv3 client that sends a synchronization packet to the server every 60 minutes. This means the GSR will attempt to set its own clock against the server once every hour. The synchronization interval as well as the NTP version number can be changed.

Note: To ensure that NTP has the correct time, you need to specify the time zone, as well. You can set the time zone by using the **system set timezone** command. When specifying daylight saving time, you'll need to use the **system set daylight-saving** command.

To configure the GSR's NTP client to synchronize its clock, enter the following command in Configure mode:

Instruct GSR's NTP server to periodically synchronize clock	**ntp set server** <*host*> **[interval** <*minutes*>**]** **[source** <*ipaddr*>**]** **[version** <*num*>**]**

Configuring the GSR 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 GSR accepts SNMP sets and gets from an SNMP manager. You can configure GSR SNMP parameters including community strings and trap server target addresses.

To configure the GSR 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 GSR allows you to configure up to three Domain Name Service (DNS) servers.

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

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

Connecting Between the GSR and Other Systems

To test a connection between the GSR and an IP host, enter the following command in User or Enable mode:

Test connection between the GSR and an IP host.	**ping** <*hostname-or-IPaddr*> **packets** <*num*> **size** <*num*> **wait** <*num*> **[flood] [dontroute]**

To open a Telnet session from the GSR to an IP host, enter the following command in User or Enable mode:

Telnet to a specified IP host.	**telnet** <*hostname-or-IPaddr*> **[socket** <*socket-number*>**]**

The GSR accepts up to four telnet sessions. You can immediately end a particular telnet session (for example, an unauthorized user is logged in to the GSR).

To end a user's telnet session, first determine the session ID by entering the following command in Enable mode:

Show current Telnet sessions.	**system show users**

To end the telnet session, enter the following command in Enable mode:

Kill the Telnet session.	**system kill telnet-session <***session-id***>**

Configuring Logging

During operation, the GSR system software sends messages to the management console. These messages include informational, warning, error, and fatal messages. Console messages can also be sent to a syslog server.

To configure a Syslog server, enter the following command in Configure mode:

Configure a Syslog server.	**system set syslog** [**server** <*hostname-or-IPaddr*>][**level** <*level-type*>][**facility** <*facility-type*>][**source** <*source-IPaddr*>][**buffer-size** <*size*>]

If a syslog server is identified and ACL logging is enabled, then messages about whether packets are forwarded or dropped because of ACL are sent to the Syslog server. Chapter 18, "Security Configuration Guide," describes ACL logging.

Monitoring Configuration

The GSR 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.

Task	Command
Display history buffer.	**cli show history**
Show terminal settings.	**cli show terminal**
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 boot PROM parameters for TFTP downloading of the system image.	**system show bootprom**

Task	Command
Show the most recent Syslog messages kept in the local syslog message buffer.	`system show syslog buffer`
Show usage information about various system resources.	`system show capacity all|chassis|task|cpu|memory`
Show the contact information (administrator name, phone number, and so on).	`system show contact`
Shows the percentage of the CPU that is currently being used.	`system show cpu-utilization`
Show the GSR date and time.	`system show date`
Show the IP addresses and domain names for DNS servers.	`system show dns`
Show environmental information, such as temperature and power supply status.	`system show environmental`
Show GSR hardware information.	`system show hardware`
Show the GSR's location.	`system show location`
Show the GSR login banner.	`system show login-banner`
Show the GSR 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 status of the switching fabric module.	`system show switching-fabric`
Show the IP address of the SYSLOG server and the level of messages the GSR sends to the server.	`system show syslog`
Lists the last five Telnet connections to the GSR.	`system show telnet-access`
Show the default terminal settings (number of rows, number of columns, and baud rate.	`system show terminal`
Show the time zone offset from UCT in minutes.	`system show timezone`

Task	Command
Show GSR uptime.	**system show uptime**
Show the current Telnet connections to the GSR.	**system show users**
Show the software version running on the GSR.	**system show version**

Chapter 2

Hot Swapping Line Cards and Control Modules

Hot Swapping Overview

This chapter describes the hot swapping functionality of the GSR. Hot swapping is the ability to replace a line card or Control Module while the GSR is operating. Hot swapping allows you to remove or install line cards without switching off or rebooting the GSR. Swapped-in line cards are recognized by the GSR and begin functioning immediately after they are installed.

On the GSR-8 and GSR-16, you can hot swap line cards and secondary control modules. On the GSR-16, you can also hot swap the secondary switching fabric module.

This chapter provides instructions for the following tasks:

- Hot swapping line cards
- Hot swapping secondary Control Modules
- Hot swapping the secondary Switching Fabric Module (GSR-16 only)

Hot Swapping Line Cards

The procedure for hot swapping a line card consists of deactivating the line card, removing it from its slot in the GSR chassis, and installing a new line card in the slot.

Deactivating the Line Card

To deactivate the line card, do one of the following:

- Press the Hot Swap button on the line card. The Hot Swap button is recessed in the line card's front panel. Use a pen or similar object to reach it.

  When you press the Hot Swap button, the Offline LED lights. Figure 1 shows the location of the Offline LED and Hot Swap button on a 1000Base-SX line card.

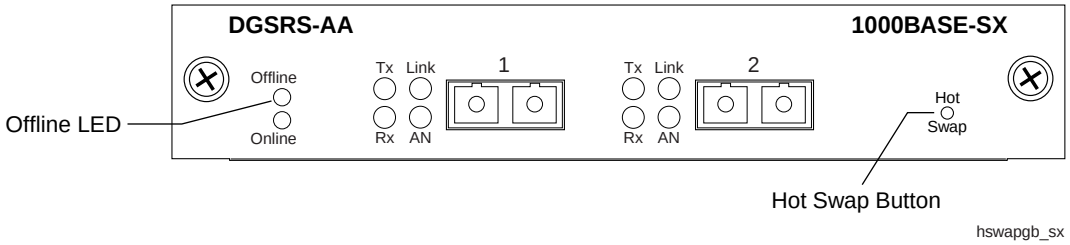


Figure 1. Location of Offline LED and Hot Swap button on a 1000Base-SX line card

- Use the **system hotswap out** command in the CLI. For example, to deactivate the line card in slot 7, enter the following command in Enable mode:

```
gs/r# system hotswap out slot 7
```

  After you enter this command, the Offline LED on the line card lights, and messages appear on the console indicating the ports on the line card are inoperative.

Note: If you have deactivated a line card and want to activate it again, simply pull it from its slot and push it back in again. (Make sure the Offline LED is lit before you pull out the line card.) The line card is activated automatically.

Alternately, if you have not removed a line card you deactivated with the **system hotswap out** command, you can reactivate it with the **system hotswap in** command. For example, to reactivate a line card in slot 7, enter the following command in Enable mode:

```
gs/r# system hotswap in slot 7
```

Removing the Line Card

To remove a line card from the GSR:

1. Make sure the Offline LED on the line card is lit.

Warning: Do not remove the line card unless the Offline LED is lit. Doing so can cause the GSR to crash.

2. Loosen the captive screws on each side of the line card.
3. Carefully remove the line card from its slot in the GSR chassis.

Installing a New Line Card

To install a new line card:

1. Slide the line card all the way into the slot, firmly but gently pressing the line card fully in place to ensure that the pins on the back of the line card are completely seated in the backplane.

   Note: Make sure the circuit card (and not the metal plate) is between the card guides. Check both the upper and lower tracks.

2. Tighten the captive screws on each side of the line card to secure it to the chassis.

   Once the line card is installed, the GSR recognizes and activates it. The Online LED button lights.

Hot Swapping One Type of Line Card With Another

You can hot swap one type of line card with another type. For example, you can replace a 10/100Base-TX line card with a 1000Base-SX line card. The GSR can be configured to accommodate whichever line card is installed in the slot. When one line card is installed, configuration statements for that line card are used; when you remove the line card from the slot and replace it with a different type, configuration statements for the new line card take effect.

To set this up, you include configuration statements for **both** line cards in the GSR configuration file. The GSR determines which line card is installed in the slot and uses the appropriate configuration statements.

For example, you may have a GSR with a 10/100Base-TX line card in slot 7 and want to hot swap it with a 1000Base-SX line card. If you include statements for both line cards in the GSR configuration file, the statements for the 1000Base-SX take effect immediately after you install it in slot 7.

Hot Swapping a Secondary Control Module

If you have a secondary control module installed on the GSR, you can hot swap it with another Control Module or line card.

Warning: You can only hot swap an **inactive** Control Module. You should never remove the active Control Module from the GSR. Doing so will crash the system.

The procedure for hot swapping a control module is similar to the procedure for hot swapping a line card. You must deactivate the Control Module, remove it from the GSR, and insert another Control Module or line card in the slot.

Deactivating the Control Module

To deactivate the Control Module:

1. Determine which is the secondary Control Module.

   Control Modules can reside in slot CM or slot CM/1 on the GSR. Usually slot CM contains the primary Control Module, and slot CM/1 contains the secondary Control Module. On the primary Control Module, the Online LED is lit, and on the secondary Control Module, the Offline LED is lit.

   Note: The Offline LED on the Control Module has a different function from the Offline LED on a line card. On a line card, it means that the line card has been deactivated. On a Control Module, a lit Offline LED means that it is standing by to take over as the primary Control Module if necessary; it **does not** mean that the Control Module has been deactivated.

2. Press the Hot Swap button on the secondary Control Module.

   When you press the Hot Swap button, all the LEDs on the Control Module (including the Offline LED) are deactivated. Figure 2 shows the location of the Offline LED and Hot Swap button on a Control Module.

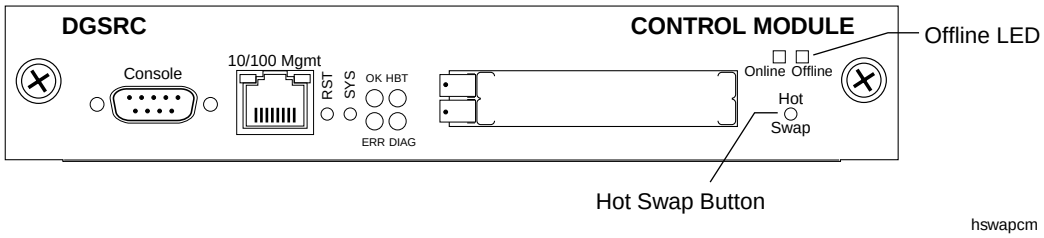


Figure 2. Location of Offline LED and Hot Swap button on a Control Module

Removing the Control Module

To remove a Control Module from the GSR:

1. Make sure that **none** of the LEDs on the Control Module are lit.
2. Loosen the captive screws on each side of the Control Module.
3. Carefully remove the Control Module from its slot in the GSR chassis.

Installing the Control Module

To install a new Control Module or line card into the slot:

Note: You can install either a line card or a Control Module in slot CM/1, but you can install **only** a Control Module in slot CM.

1. Slide the Control Module or line card all the way into the slot, firmly but gently pressing it fully in place to ensure that the pins on the back of the card are completely seated in the backplane.

   Note: Make sure the circuit card (and not the metal plate) is between the card guides. Check both the upper and lower tracks.

2. Tighten the captive screws on each side of the Control Module or line card to secure it to the chassis.

   On a line card, the Online LED lights, indicating it is now active.

   On a secondary Control Module, the Offline LED lights, indicating it is standing by to take over as the primary Control Module if necessary.

Hot Swapping a Switching Fabric Module (GSR-16 only)

The GSR-16 has slots for two Switching Fabric Modules. While the GSR-16 is operating, you can install a second Switching Fabric Module. If two Switching Fabric Modules are installed, you can hot swap one of them.

When you remove one of the Switching Fabric Modules, the other goes online and stays online until it is removed or the GSR-16 is powered off. When the GSR-16 is powered on again, the Switching Fabric Module in slot "Fabric 1", if one is installed there, becomes the active Switching Fabric Module.

Warning: You can only hot swap a Switching Fabric Module if two are installed on the GSR-16. If only one Switching Fabric Module is installed, and you remove it, the GSR-16 will crash.

The procedure for hot swapping a Switching Fabric Module is similar to the procedure for hot swapping a line card or Control Module. You deactivate the Switching Fabric Module, remove it from the GSR, and insert another Switching Fabric Module in the slot.

Note: You cannot deactivate the Switching Fabric Module with the **system hotswap** command.

To deactivate the Switching Fabric Module:

1. Press the Hot Swap button on the Switching Fabric Module you want to deactivate.

   The Online LED goes out and the Offline LED lights. Figure 3 shows the location of the Offline LED and Hot Swap button on a Switching Fabric Module.

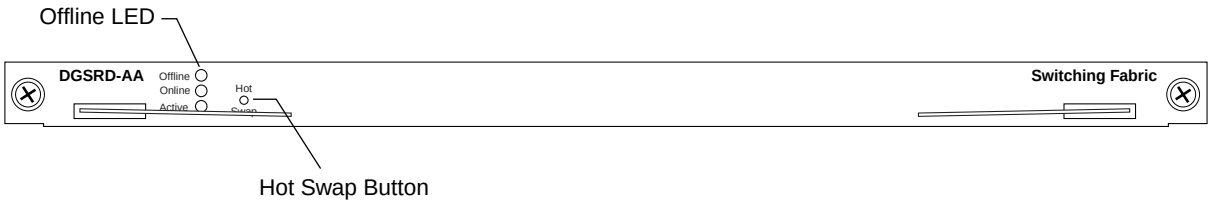


Figure 3. Location of Offline LED and Hot Swap button on a Switching Fabric Module

To remove the Switching Fabric Module:

1. Loosen the captive screws on each side of the Switching Fabric Module.
2. Pull the metal tabs on the Switching Fabric Module to free it from the connectors holding it in place in the chassis.
3. Carefully remove the Switching Fabric Module from its slot.

To install a Switching Fabric Module:

1. Slide the Switching Fabric Module all the way into the slot, firmly but gently pressing to ensure that the pins on the back of the module are completely seated in the backplane.

   Note: Make sure the circuit card (and not the metal plate) is between the card guides. Check both the upper and lower tracks.

2. Tighten the captive screws on each side of the Switching Fabric Module to secure it to the chassis.

Chapter 3

Bridging Configuration Guide

Bridging Overview

The DIGITAL GIGAswitch/Router provides the following bridging functions:

- Compliance with the IEEE 802.1d standard
- Compliance with the IGMP multicast bridging standard
- Wire-speed address-based bridging or flow-based bridging
- 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®)
- Frame filtering based on MAC address for bridged and multicast traffic
- 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 GSR provides the following types of wire-speed bridging:

Address-based bridging - The GSR 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 GSR's own MAC address, the packet is routed rather than bridged.

Flow-based bridging - The GSR 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 GSR 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 GSR 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 VLAN 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.

VLANs are primarily used 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:

- Port based
- MAC address based
- Protocol based
- Subnet based
- Multicast based
- 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. The VLAN database returns the name of 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.

GSR VLAN Support

The GSR supports:

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

When using the GSR as an L2 bridge/switch, use the port-based and protocol-based VLAN types. When using the GSR 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 GSR, as seen in the section on configuring the GSR.

VLANs and the GSR

VLANs are an integral part of the GSR family of switching routers. The GSR switching routers can function as layer-2 (L2) switches as well as fully-functional layer-3 (L3) routers. Hence they 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 GSR switching routers are tightly coupled.

The GSR 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 GSR switching router to use port-based VLANs, protocol-based VLANs, or a mixture of the two types.

The GSR can also be used purely as a router, i.e., each physical port of the GSR 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 GSR 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 GSR are subnet-based VLANs.

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

Ports, VLANs, and L3 Interfaces

The term *port* refers to a physical connector on the GSR, such as an ethernet port. Each port must belong to at least one VLAN. When the GSR 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 GSR 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 a GSR 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 GSR 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 GSR A and B are both configured with VLANs V1 and V2.

Then a frame arriving at a port on GSR A must be sent to GSR B, if the frame belongs to VLAN V1 or to VLAN V2. Thus the ports on GSR A and B which connect the two GSRs 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 GSR 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 GSR when L3 interfaces are created (implicit VLANs).

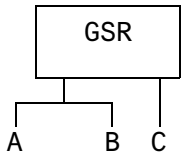
Configuring GSR Bridging Functions

Configuring Address-based or Flow-based Bridging

The GSR 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 GSR 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 a 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 is enabled compared to address-based bridging.

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

With the GSR 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 on layer-2 traffic flows.

To enable flow-based bridging on a port, 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

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

The GSR supports per VLAN spanning tree. By default, all the VLANs defined belong to the default spanning tree. You can create a separate instance of spanning tree using the following command:

Create spanning tree for a VLAN.	**pvst create spanningtree vlan-name** <*string*>

By default, spanning tree is disabled on the GSR. To enable spanning tree on the GSR, you perform the following tasks on the ports where you want spanning tree enabled..

Enable spanning tree on one or more ports for default spanning tree.	**stp enable port** <*port-list*>
Enable spanning tree on one or more ports for a particular VLAN.	**pvst enable port** <*port-list*> **spanning-tree** <*string*>

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 for default spanning tree.	**stp set bridging priority** <*num*>
Set the bridge priority for a particular instance of spanning tree.	**pvst set bridging spanning-tree** <*string*> **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 for default spanning tree.	**stp set port** <*port-list*> **priority** <*num*>
Establish a priority for a specified interface for a particular instance of spanning tree.	**pvst set port** <*port-list*> **spanning-tree** <*string*> **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 for default spanning tree.	**stp set port** <*port-list*> **port-cost** <*num*>
Set a different port cost other than the defaults for a particular instance of spanning tree.	**pvst set port** <*port-list*> **spanning-tree** <*string*> **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 for default spanning tree.	**stp set bridging hello-time** <*num*>
Specify the interval between hello time for a particular instance of spanning tree.	**pvst set bridging spanning-tree** <*string*> **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 for default spanning tree.	**stp set bridging forward-delay** *<num>*
Set the default of the forward delay interval for a particular instance of spanning tree.	**pvst set bridging spanning-tree** *<string>* **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 for default spanning tree.	**stp set bridging max-age** *<num>*
Change the amount of time a bridge will wait to hear BPDUs from the root bridge for a particular instance of spanning tree.	**pvst set bridging spanning-tree** *<string>* **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, enter the following command in Configure mode.

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

Adding Ports to a VLAN

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

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

Configuring VLAN Trunk Ports

The GSR supports standards-based VLAN trunking between multiple GSRs 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 GSRs. These VLAN IDs extend the VLAN broadcast domain to more than one GSR.

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

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

Configuring VLANs for Bridging

The GSR allows you to create VLANs for AppleTalk, DECnet®, SNA, and IPv6 traffic as well as for IP and IPX traffic. You can create a VLAN for handling traffic for a single protocol, such as a DECnet VLAN. Or, you can create a VLAN that supports several specific protocols, such as SNA and IP traffic.

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

Configuring Layer-2 Filters

Layer-2 security filters on the GSR 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 GSR 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 GSR provides display of bridging statistics and configurations contained in the GSR.

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 show**

Configuration Examples

VLANs are used to associate physical ports on the GSR 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 a VLAN and then assign ports to the VLAN. This section shows examples of creating an IP or IPX VLAN and a DECnet, SNA, and AppleTalk VLAN.

Creating an IP or IPX VLAN

In this example, servers connected to port gi.1.(1-2) on the GSR 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'.

First, create an IP VLAN named 'BLUE'

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

Next, assign ports to the 'BLUE' VLAN.

```
gs/r(config)# vlan add ports et.4.(1-8),gi.1.(1-2) to BLUE
```

Creating a non-IP/non-IPX VLAN

In this example, SNA, DECnet, and AppleTalk hosts are connected to et.1.1 and et.2.(1-4). You can associate all the ports containing these hosts to a VLAN called 'RED' with the VLAN ID 5.

First, create a VLAN named 'RED'

```
gs/r(config)# vlan create RED sna dec appletalk id 5
```

Next, assign ports to the 'RED' VLAN.

```
gs/r(config)# vlan add ports et.1.1, et.2.(1-4) to RED
```

Chapter 4

SmartTRUNK Configuration Guide

Overview

This chapter explains how to configure and monitor SmartTRUNKs on the GSR. A SmartTRUNK is DIGITAL Equipment Corporation's technology for load balancing and load sharing. For a description of the SmartTRUNK commands, see the "**smarttrunk** commands" section of the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.

On the GSR, a SmartTRUNK is a group of two or more ports that have been logically combined into a single port. Multiple physical connections between devices are aggregated into a single logical, high-speed path that acts as a single link. Traffic is balanced across all interfaces in the combined link, increasing overall available system bandwidth.

SmartTRUNKs allow administrators the ability to increase bandwidth at congestion points in the network, thus eliminating potential traffic bottlenecks. SmartTRUNKs also provide improved data link resiliency. If one port in a SmartTRUNK should fail, its load is distributed evenly among the remaining ports and the entire SmartTRUNK link remains operational.

SmartTRUNK is DIGITAL's standard for building high-performance links between DIGITAL's switching platforms. SmartTRUNKs can interoperate with switches, routers, and servers from other vendors as well as DIGITAL platforms.

SmartTRUNKs are compatible with all GSR features, including VLANs, STP, VRRP, etc. SmartTRUNK operation is supported over different media types and a variety of technologies including 10/100/1000 Mbps Ethernet.

Configuring SmartTRUNKs

To create a SmartTRUNK:

1. Create a SmartTRUNK and specify a control protocol for it.
2. Add physical ports to the SmartTRUNK.
3. Specify the policy for distributing traffic across SmartTRUNK ports. This step is optional; by default, the GSR distributes traffic to ports in a round-robin (sequential) manner.

Creating a SmartTRUNK

When you create a SmartTRUNK, you specify if the DEC® Hunt Group Control Protocol is to be used or no control protocol is to be used:

- If you are connecting the SmartTRUNK to another GSR or to other DIGITAL devices (such as the DIGITAL GIGAswitch/Router), specify the DEC Hunt Group Control Protocol. The DEC Hunt Group Control Protocol is useful in detecting errors like transmit/receive failures, misconfiguration, etc.
- If you are connecting the SmartTRUNK to a device that does not support the DEC Hunt Group Control Protocol, such as those devices that support Cisco's EtherChannel® technology, specify no control protocol. Only link failures are detected in this mode.

To create a SmartTRUNK, enter the following command in Configure mode:

Create a SmartTRUNK that will be connected to a device that supports the DEC Hunt Group Control Protocol.	**smarttrunk create <smarttrunk> protocol huntgroup**
Create a SmartTRUNK that will be connected to a device that does not support the DEC Hunt Group Control Protocol.	**smarttrunk create <smarttrunk> protocol no-protocol**

Add Physical Ports to the SmartTRUNK

You can add any number of ports to a SmartTRUNK. The limit is the number of ports on the GSR. Any port on any module can be part of a SmartTRUNK. If one module should go down, the remaining ports on other modules will remain operational.

Ports added to a SmartTRUNK must:

- Be set to full duplex.
- Be in the same VLAN.
- Have the same properties (L2 aging, STP state, and so on).

To add ports to a SmartTRUNK, enter the following command in Configure mode:

Create a SmartTRUNK that will be connected to a device that supports the DEC Hunt Group Control Protocol.	**smarttrunk add ports** *<port list>* **to** <**smarttrunk**>

Specify Traffic Distribution Policy (Optional)

The default policy for distributing traffic across the ports in a SmartTRUNK is "round-robin," where the GSR selects the port on a rotating basis. The other policy that can be chosen is "link-utilization," where packets are sent to the least-used port in a SmartTRUNK. You can choose to specify the link-utilization policy for a particular SmartTRUNK, a list of SmartTRUNKs, or for all SmartTRUNKs on the GSR.

Specify traffic distribution policy.	**smarttrunk set load-policy on** <**smarttrunk** *list*>**|all-smarttrunks round-robin|link-utilization**

Monitoring SmartTRUNKs

Statistics are gathered for data flowing through a SmartTRUNK and each port in the SmartTRUNK.

To display SmartTRUNK statistics, enter one of the following commands in Enable mode:.

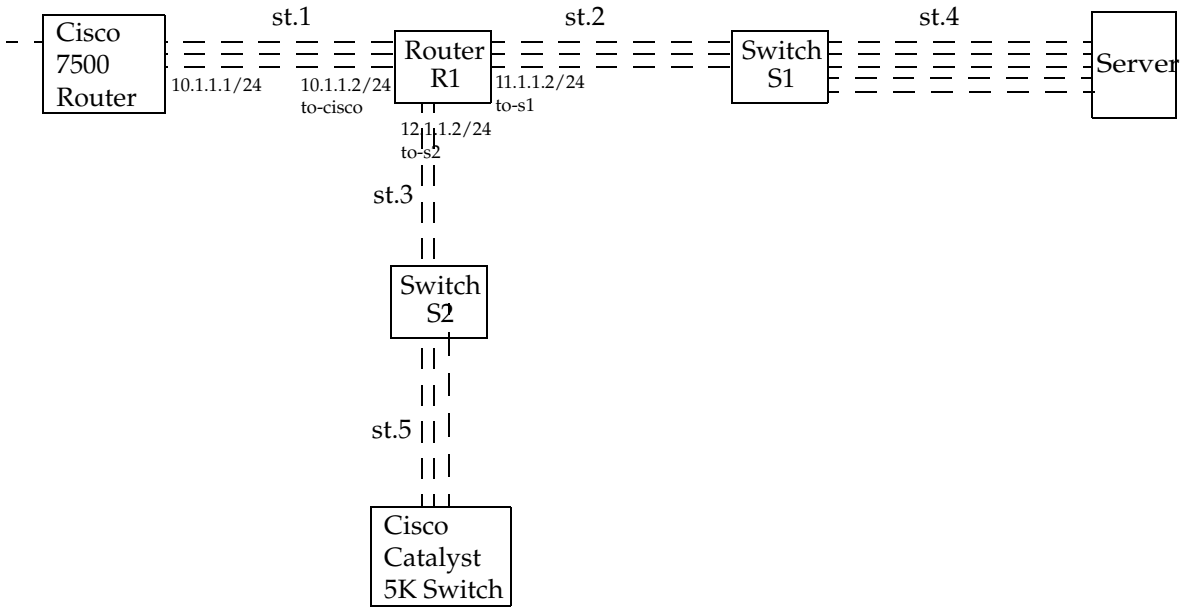
Display information about all SmartTRUNKs and the control protocol used.	**smarttrunk show trunks**
Display statistics on traffic distribution on SmartTRUNK	**smarttrunk show distribution** <**smarttrunk** *list*>**|all-smarttrunks**
Display information about the control protocol on a SmartTRUNK.	**smarttrunk show protocol-state** <**smarttrunk** *list*>**|all-smarttrunks**
Display information about the SmartTRUNK connection (DEC Hunt Group Control Protocol connections only).	**smarttrunk show connections** <**smarttrunk** *list*>**|all-smarttrunks**

To clear statistics for SmartTRUNK ports, enter the following command in Enable mode:.

Clear load distribution statistics for SmartTRUNK ports.	**smarttrunk clear load-distribution** <**smarttrunk** *list*>**|all-smarttrunk**

Example Configurations

The following shows a network design based on SmartTRUNKs. R1 is an GSR operating as a router, while S1 and S2 are GSRs operating as switches.



The following is the configuration for the Cisco 7500 router:

```
interface port-channel 1
ip address 10.1.1.1 255.255.255.0
ip route-cache distributed
interface fasteth 0/0
no ip address
channel-group 1
```

The following is the configuration for the Cisco Catalyst 5K switch:

```
set port channel 3/1-2 on
```

The following is the SmartTRUNK configuration for the GSR labeled 'R1' in the diagram:

```
smarttrunk create st.1 protocol no-protocol
smarttrunk create st.2 protocol huntgroup
smarttrunk create st.3 protocol huntgroup
smarttrunk add ports et.1(1-2) to st.1
smarttrunk add ports et.2(1-2) to st.2
smarttrunk add ports et.3(1-2) to st.3
interface create ip to-cisco address-netmask 10.1.1.2/24 port st.1
interface create ip to-s1 address-netmask 11.1.1.2/24 port st.2
interface create ip to-s2 address-netmask 12.1.1.2/24 port st.3
```

The following is the SmartTRUNK configuration for the GSR labeled 'S1' in the diagram:

```
smarttrunk create st.2 protocol huntgroup
smarttrunk create st.4 protocol no-protocol
smarttrunk add ports et.1(1-2) to st.2
smarttrunk add ports et.2(1-2) to st.4
```

The following is the SmartTRUNK configuration for the GSR labeled 'S2' in the diagram:

```
smarttrunk create st.3 protocol huntgroup
smarttrunk create st.5 protocol no-protocol
smarttrunk add ports et.1(1-2) to st.3
smarttrunk add ports et.2(1-2) to st.5
```

Chapter 5

DHCP Configuration Guide

DHCP Overview

The Dynamic Host Configuration Protocol (DHCP) server on the GSR provides dynamic address assignment and configuration to DHCP capable end-user systems, such as Microsoft® Windows® 95/98, Microsoft Windows NT, and Apple® Macintosh® systems. You can configure the server to provide a dynamic IP address from a pre-allocated pool of IP addresses or a static IP address. You can also configure parameters for use by the clients, such as default gateway and network masks, and system-specific parameters, such as NetBIOS Name Server and NetBIOS node type of the client.

The amount of time that a particular IP address is valid for a system is called a *lease*. The GSR maintains a *lease database* which contains information about each assigned IP address, the MAC address to which it is assigned, the lease expiration, and whether the address assignment is dynamic or static. The DHCP lease database is stored in flash memory and can be backed up on a remote TFTP or RCP server. You can configure the intervals at which updates to the lease database (and backup) are done. Upon system reboot, the lease database will be loaded either from flash memory or from the TFTP or RCP server.

Note: The GSR DHCP server is not designed to work as the primary DHCP server in an enterprise environment with hundreds or thousands of clients that are constantly seeking IP address assignment or reassignment. A standalone DHCP server with a redundant backup server may be more suitable for this enterprise environment.

Configuring DHCP

By default, the DHCP server is not enabled on the GSR. You can selectively enable DHCP service on particular interfaces and not others. To enable DHCP service on an interface, you must first define a DHCP *scope*. A scope consists of a pool of IP addresses and a set of parameters for a DHCP client. The parameters are used by the client to configure its network environment, for example, the default gateway and DNS domain name.

To configure DHCP on the GSR, you must configure an IP address pool, client parameters, and optional static IP address for a specified scope. Where several subnets are accessed through a single port, you can also define multiple scopes on the same interface and group the scopes together into a "superscope."

Configuring an IP Address Pool

To define a pool of IP addresses that the DHCP server can assign to a client, enter the following command in Configure mode:

Define pool of IP addresses to be used by clients.	**dhcp** <*scope*> **define pool** <*ip-range*>

Configuring Client Parameters

You can configure the client parameters shown in the table below.

Table 3. Client Parameters

Parameter	Value
address-mask	Address/netmask of the scope's subnet (This parameter is *required* and must be defined *before* any other client parameters are specified.)
broadcast	Broadcast address
bootfile	Client boot file name
dns-domain	DNS domain name
dns-server	IP address of DNS server
gateway	IP address of default gateway
lease-time	Amount of time the assigned IP address is valid for the system

Table 3. Client Parameters

Parameter	Value
netbios-name-server	IP address of NetBIOS Name Server (WINS server)
netbios-node-type	NetBIOS node type of the client
netbios-scope	NetBIOS scope of the client

To define the parameters that the DHCP server gives the clients, enter the following command in Configure mode:

Define client parameters.	**dhcp** <*scope*> **define parameters** <*parameter*> <*value*>...

Configuring a Static IP Address

To define a static IP address that the DHCP server can assign to a client with a specific MAC address, enter the following command in Configure mode:

Define static IP address for a particular MAC address.	**dhcp** <*scope*> **define static-ip** <*ipaddr*> **mac-address** <*macaddr*> [<*parameter*> <*value*>...]

Grouping Scopes with a Common Interface

You can apply several scopes to the same physical interface. For example, scopes can define address pools on different subnets that all are accessed through the same GSR port. In this case, scopes that use the same interface must be grouped together into a "superscope."

To attach a scope to a superscope, enter the following command in Configure mode:

Attach a scope to a superscope.	**dhcp** <*scope*> **attach superscope** <*name*>

Configuring DHCP Server Parameters

You can configure several "global" parameters that affect the behavior of the DHCP server itself.

To configure global DHCP server parameters, enter the following commands in Configure mode:

Specify a remote location to back up the lease database.	**dhcp global set lease-database** *<url>*
Specify the intervals at which the lease database is updated.	**dhcp global set commit-interval** *<hours>*

Updating the Lease Database

After each client transaction, the DHCP server does not immediately update the information in the lease database. Lease update information is stored in flash memory and flushed to the database at certain intervals. You can use the **dhcp global set commit-interval** command to specify this interval; the default is one hour.

To force the DHCP server to immediately update its lease database, enter the following command in Enable mode:

Force the server to update its lease database.	**dhcp flush**

Monitoring the DHCP Server

To display information from the lease database:

Show lease database information.	**dhcp show binding [active|expired|static]**

To display the number of allocated bindings for the DHCP server and the maximum number allowed:

Show the number of allocated bindings for the DHCP server.	**dhcp show num-clients**

DHCP Configuration Examples

The following configuration describes DHCP configuration for a simple network with just one interface on which DHCP service is enabled to provide both dynamic and static IP addresses.

1. Create an IP VLAN called ‘client_vlan’.

```
vlan create client_vlan ip
```

2. Add all Fast Ethernet ports in the GSR to the VLAN ‘client_vlan’.

```
vlan add port et.*.* to client_vlan
```

3. Create an IP interface called ‘clients’ with the address 10.1.1.1 for the VLAN ‘client_vlan’.

```
interface create ip clients address-netmask 10.1.1.1./16 vlan
    client_vlan
```

4. Define DHCP network parameters for the scope ‘scope1’.

```
dhcp scope1 define parameters address-netmask 10.1.0.0/16 gateway
    10.1.1.1 lease-time 720 dns-domain acme.com dns-server
    10.2.45.67 netbios-name-server 10.1.55.60
```

5. Define an IP address pool for addresses 10.1.1.10 through 10.1.1.20.

```
dhcp scope1 define pool 10.1.1.10-10.1.1.20
```

6. Define another IP address pool for addresses 10.1.1.40 through 10.1.1.50.

```
dhcp scope1 define pool 10.1.1.40-10.1.1.50
```

7. Define a static IP address for 10.1.7.5.

```
dhcp scope1 define static-ip 10.1.7.5 mac-address 08:00:20:11:22:33
```

8. Define another static IP address for 10.1.7.7. and give it a specific gateway address of 10.1.1.2.

```
dhcp scope1 define static-ip 10.1.7.7 mac-address
    08:00:20:aa:bb:cc:dd gateway 10.1.1.2
```

9. Specify a remote lease database on the TFTP server 10.1.89.88.

```
dhcp global set lease-database tftp://10.1.89.88/lease.db
```

10. Specify a database update interval of every 15 minutes.

```
dhcp global set commit-interval 15
```

Configuring Secondary Subnets

In some network environments, multiple logical subnets can be imposed on a single physical segment. These logical subnets are sometimes referred to as "secondary subnets" or "secondary networks." For these environments, the DHCP server may need to give out addresses on different subnets. The DNS server, DNS domain, and WINS server may be the same for clients on different secondary subnets, however, the default gateway will most likely be different since it must be a router on the client's local subnet.

The following example shows a simple configuration to support secondary subnets 10.1.x.x and 10.2.x.x.

1. Define the network parameters for 'scope1' with the default gateway 10.1.1.1.

```
dhcp scope1 define parameters address-netmask 10.1.0.0/16 gateway
    10.1.1.1 dns-domain acme.com dns-server 10.1.44.55
```

2. Define the address pool for 'scope1'.

```
dhcp scope1 define pool 10.1.1.10-10.1.1.20
```

3. Define the network parameters for 'scope2' with the default gateway 10.2.1.1.

```
dhcp scope2 define parameters address-netmask 10.2.0.0/16 gateway
    10.2.1.1 dns-domain acme.com dns-server 10.1.77.88
```

4. Define the address pool for 'scope2'.

```
dhcp scope2 define pool 10.2.1.40-10.2.1.50
```

5. Create a superscope 'super1' that includes 'scope1'.

```
dhcp scope1 attach superscope super1
```

6. Include ‘scope2’ in the superscope ‘super1’.

```
dhcp scope2 attach superscope super1
```

Since there are multiple pools of IP addresses, the pool associated with ‘scope1’ is used first since ‘scope1’ is applied to the interface before ‘scope2’. Clients that are given an address from ‘scope1’ will also be given parameters from ‘scope1,’ which includes the default gateway 10.1.1.1 that resides on the 10.1.x.x subnet. When all the addresses for ‘scope1’ are assigned, the server will start giving out addresses from ‘scope2’ which will include the default gateway parameter 10.2.1.1 on subnet 10.2.x.x.

Secondary Subnets and Directly-Connected Clients

A directly-connected client is a system that resides on the same physical network as the DHCP server and does not have to go through a router or relay agent to communicate with the server. If you configure the DHCP server on the GSR to service directly-connected clients on a secondary subnet, you must configure the secondary subnet using the **interface add ip** command. The **interface add ip** command configures a secondary address for an interface that was previously created with the **interface create ip** command.

The following example shows a simple configuration to support directly-connected clients on a secondary subnet.

1. Create an interface ‘clients’ with the primary address 10.1.1.1.

```
interface create ip clients address-mask 10.1.1.1/16 port et.1.1
```

2. Assign a secondary address 10.2.1.1 to the interface ‘clients’.

```
interface add ip clients address-mask 10.2.1.1/16
```

3. Define the network parameters for ‘scope1’ with the default gateway 10.1.1.1.

```
dhcp scope1 define parameters address-netmask 10.1.0.0/16 gateway
    10.1.1.1 dns-domain acme.com dns-server 10.1.44.55
```

4. Define the address pool for ‘scope1’.

```
dhcp scope1 define pool 10.1.1.10-10.1.1.20
```

5. Define the network parameters for ‘scope2’ with the default gateway 10.2.1.1.

```
dhcp scope2 define parameters address-netmask 10.2.0.0/16 gateway
    10.2.1.1 dns-domain acme.com dns-server 10.1.77.88
```

6. Define the address pool for ‘scope2’.

```
dhcp scope2 define pool 10.2.1.40-10.2.1.50
```

7. Create a superscope ‘super1’ that includes ‘scope1’.

```
dhcp scope1 attach superscope super1
```

8. Include ‘scope2’ in the superscope ‘super1’.

```
dhcp scope2 attach superscope super1
```

For clients on the secondary subnet, the default gateway is 10.2.1.1, which is also the secondary address for the interface ‘clients’.

Interacting with Relay Agents

For clients that are not directly connected to the DHCP server, a relay agent (typically a router) is needed to communicate between the client and the server. The relay agent is usually only needed during the initial leasing of an IP address. Once the client obtains an IP address and can connect to the network, the renewal of the lease is performed between the client and server without the help of the relay agent.

The default gateway for the client must be capable of reaching the GSR’s DHCP server. The GSR must also be capable of reaching the client’s network. The route must be configured (with static routes, for example) or learned (with RIP or OSPF, for example) so that the DHCP server can reach the client.

The following example shows a simple configuration to support clients across a relay agent.

1. Create an interface ‘clients’ with the primary address 10.1.1.1.

```
interface create ip clients address-mask 10.1.1.1/16 port et.3.3
```

2. Define a static route to the 10.5.x.x. subnet using the gateway 10.1.7.10 which tells the DHCP server how to send packets to the client on the 10.5.x.x subnet.

```
ip add route 10.5.0.0/16 gateway 10.1.7.10
```

3. Define the network parameters for ‘scope1’ with the default gateway 10.5.1.1 (the relay agent for the client).

```
dhcp scope1 define parameters address-netmask 10.5.0.0/16 gateway
    10.5.1.1 dns-domain acme.com
```

4. Define the address pool for ‘scope1’.

```
dhcp scope1 define pool 10.5.1.10-10.5.1.20
```

Chapter 6

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 GSR supports standards-based TCP, UDP, and IP.

IP Routing Protocols

The GSR supports standards-based unicast and multicast routing. Unicast routing protocol support includes 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 GSR 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 GSR 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 GSR 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 GSR 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 GSR 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 (ARP)

The GSR 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 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 GSR can be configured for proxy ARP. The GSR 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 GSR will respond to ARP requests from a host with a ARP reply packet containing the GSR MAC address. Proxy ARP is enabled by default on the GSR.

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

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

Configuring Reverse Address Resolution Protocol (RARP)

Reverse Address Resolution Protocol (RARP) works exactly the opposite of ARP. Taking a MAC address as input, RARP determines the associated IP address. RARP is useful for X-terminals and diskless workstations that may not have an IP address when they boot. They can submit their MAC address to a RARP server on the GSR, which returns an IP address.

Configuring RARP on the GSR consists of two steps:

- Letting the GSR know which IP interfaces to respond to
- Defining the mappings of MAC addresses to IP addresses

Specifying IP Interfaces for RARP

To specify the interfaces that the RARP server on the GSR should respond to, enter the following command in Configure mode:

Specify interfaces for RARP.	**rarpd set interface** <*InterfaceName*>**|all**

Defining MAC-to-IP Address Mappings

To map a MAC address to an IP address, enter the following command in Configure mode:

Map a MAC address to an IP address.	**rarpd add hardware-address** <*MAC-addr*> **ip-address** <*IPaddr*>

There is no limit to the number of address mappings you can configure.

Optionally, you can create a list of mappings with a text editor and then use TFTP to upload the text file to the GSR. The format of the text file must be as follows:

```
MAC-address1 IP-address1
MAC-address2 IP-address2
...
MAC-addressn IP-addressn
```

Then place the text file on a TFTP server that the GSR can access and enter the following command in Enable mode:

```
gs/r# copy tftp-server to ethers
TFTP server? <IPaddr-of-TFTP-server>
Source filename? <filename>
```

Monitoring RARP

You can use the following commands to obtain information about the GSR's RARP configuration:

Display the interfaces to which the RARP server responds.	**rarpd show interface**
Display the existing MAC-to-IP address mappings	**rarpd show mappings**
Display RARP statistics.	**statistics show rarp** <*InterfaceName*>**|all**

Configuring DNS Parameters

The GSR 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 GSR. The domain name is used by the GSR 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 GSR 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 GSR, 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]**

Configuring IP Helper

You can configure the GSR to forward UDP broadcast packets received on a given interface to all other interfaces or to a specified IP address. You can specify a UDP port number for which UDP broadcast packets with that destination port number will be forwarded. By default, if no UDP port number is specified, the GSR will forward UDP broadcast packets for the following six services:

- BOOTP/DHCP (port 67 and 68)
- DNS (port 37)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)
- TACACS Server (port 49)
- Time Service (port 37)

To configure a destination to which UDP packets will be forwarded, enter the following command in Configure mode:

Specify local subnet interface, destination "helper" IP address, and UDP port number to forward.	**ip helper-address interface** <*interface-name*> <*helper-address*> | **all-interfaces** [<*udp-port#*>]

Configuring Direct Broadcast

You can configure the GSR to forward all directed broadcast traffic from the local subnet to a specified IP address or all associated IP addresses. This is a more efficient method than defining only one local interface and remote IP address destination at a time with the **ip-helper** command when you are forwarding traffic from more than one interface in the local subnet to a remote destination IP address.

To forward all directed broadcast traffic to a specified IP address, enter the following command in Configure mode:

Forward directed broadcast traffic.	**ip enable directed-broadcast interface** *<interface name>***|all**

Configuring Denial of Service (DOS)

By default, the GSR installs flows in the hardware so that packets sent as directed broadcasts are dropped in hardware, if directed broadcast is not enabled on the interface where the packet is received. You can disable this feature, causing directed broadcast packets to be processed on the GSR even if directed broadcast is not enabled on the interface receiving the packet.

Similarly, the GSR installs flows to drop packets destined for the GSR for which service is not provided by the GSR. This prevents packets for unknown services from slowing the CPU. You can disable this behavior, causing these packets to be processed by the CPU.:

Disables the directed-broadcast-protection feature of the GSR.	**ip dos disable directed-broadcast-protection**
Disables the port-attack-protection feature of the GSR.	**ip dos disable port-attack-protection**

Monitoring IP Parameters

The GSR 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 all**
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**

Configuring Router Discovery

The router discovery server on the GSR periodically sends out router advertisements to announce the existence of the GSR to other hosts. The router advertisements are multicast or broadcast to each interface on the GSR on which it is enabled and contain a list of the addresses on the interface and the preference of each address for use as a default route for the interface. A host can also send a router solicitation, to which the router discovery server on the GSR will respond with a unicast router advertisement.

On systems that support IP multicasting, router advertisements are sent to the 'all-hosts' multicast address 224.0.0.1 by default. You can specify that broadcast be used, even if IP multicasting is available. When router advertisements are sent to the all-hosts multicast address or an interface is configured for the limited broadcast address 255.255.255.255, the router advertisement includes all IP addresses configured on the physical interface. When router advertisements are sent to a net or subnet broadcast, then only the address associated with the net or subnet is included.

To start and stop router discovery on the GSR, enter the following commands in Configure mode:

Start router discovery.	**rdisc start**
Stop router discovery.	**rdisc stop**

To configure router advertisement, enter the following commands in Configure mode:

Define IP address to be included in router advertisements.	**rdisc add address** <*hostname-or-ipaddr*>
Enable router advertisement on an interface.	**rdisc add interface** <*interface name*>**|all**
Configure router advertisement for a specific address.	**rdisc set address** <*ipaddr*> **type multicast|broadcast** **advertise enable|disable** **preference** <*number*>**|ineligible**
Configure router advertisement for an interface or all interfaces.	**rdisc set interface** <*name*>**|all min-adv-interval** <*number*> **max-adv-interval** <*number*> **lifetime** <*number*>

To show the state of router discovery on the GSR, enter the following command in Enable mode:

Show router discovery state.	**rdisc show all**

Configuration Examples

Assigning IP/IPX Interfaces

To enable routing on the GSR, you must assign an IP or IPX interface to a VLAN. To assign an IP or IPX interface named 'RED' to the 'BLUE' VLAN, enter the following command:

```
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 7

VRRP Configuration Guide

VRRP Overview

This chapter explains how to set up and monitor the Virtual Router Redundancy Protocol (VRRP) on the GSR. VRRP is defined in RFC 2338.

End host systems on a LAN are often configured to send packets to a statically configured default router. If this default router becomes unavailable, all the hosts that use it as their first hop router become isolated on the network. VRRP provides a way to ensure the availability of an end host's default router.

This is done by assigning IP addresses that end hosts use as their default route to a "virtual router." A Master router is assigned to forward traffic designated for the virtual router. If the Master router should become unavailable, a backup router takes over and begins forwarding traffic for the virtual router. As long as one of the routers in a VRRP configuration is up, the IP addresses assigned to the virtual router are always available, and the end hosts can send packets to these IP addresses without interruption.

Configuring VRRP

This section presents three sample VRRP configurations:

- A basic VRRP configuration with one virtual router
- A symmetrical VRRP configuration with two virtual routers
- A multi-backup VRRP configuration with three virtual routers

Basic VRRP Configuration

Figure 4 shows a basic VRRP configuration with a single virtual router. Routers R1 and R2 are both configured with one virtual router (VRID=1). Router R1 serves as the Master and Router R2 serves as the Backup. The four end hosts are configured to use 10.0.0.1/16 as the default route. IP address 10.0.0.1/16 is associated with virtual router VRID=1.

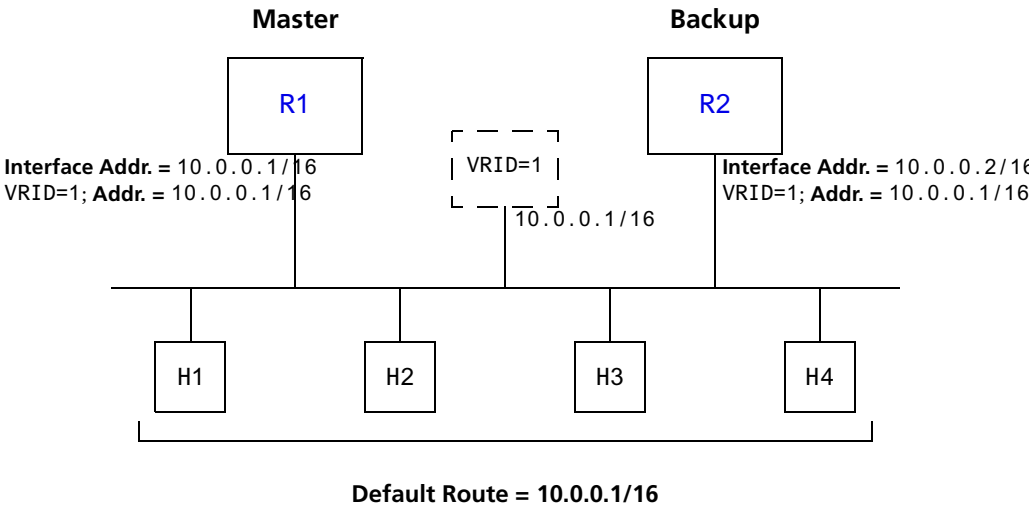


Figure 4. Basic VRRP Configuration

If Router R1 should become unavailable, Router R2 would take over virtual router VRID=1 and its associated IP addresses. Packets sent to 10.0.0.1/16 would go to Router R2. When Router R1 comes up again, it would take over as Master, and Router R2 would revert to Backup.

Configuration of Router R1

The following is the configuration file for Router R1 in Figure 4.

```
1: interface create ip test address-netmask 10.0.0.1/16 port et.1.1
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
4: ip-redundancy start vrrp 1 interface test
```

Line 1 adds IP address 10.0.0.1/16 to interface test, making Router R1 the owner of this IP address. Line 2 creates virtual router `VRID=1` on interface test. Line 3 associates IP address 10.0.0.1/16 with virtual router `VRID=1`. Line 4 starts VRRP on interface test.

In VRRP, the router that owns the IP address associated with the virtual router is the Master. Any other routers that participate in this virtual router are Backups. In this configuration, Router R1 is the Master for virtual router `VRID=1` because it owns 10.0.0.1/16, the IP address associated with virtual router `VRID=1`.

Configuration for Router R2

The following is the configuration file for Router R2 in Figure 4.

```
1: interface create ip test address-netmask 10.0.0.2/16 port et.1.1
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
4: ip-redundancy start vrrp 1 interface test
```

The configuration for Router R2 is nearly identical to Router R1. The difference is that Router R2 does not own IP address 10.0.0.1/16. Since Router R2 does not own this IP address, it is the Backup. It will take over from the Master if it should become unavailable.

Symmetrical Configuration

Figure 5 shows a VRRP configuration with two routers and two virtual routers. Routers R1 and R2 are both configured with two virtual routers (`VRID=1` and `VRID=2`).

Router R1 serves as:

- Master for `VRID=1`
- Backup for `VRID=2`

Router R2 serves as:

- Master for `VRID=2`
- Backup for `VRID=1`

This configuration allows you to load-balance traffic coming from the hosts on the 10.0.0.0/16 subnet and provides a redundant path to either virtual router.

Note: This is the recommended configuration on a network using VRRP.

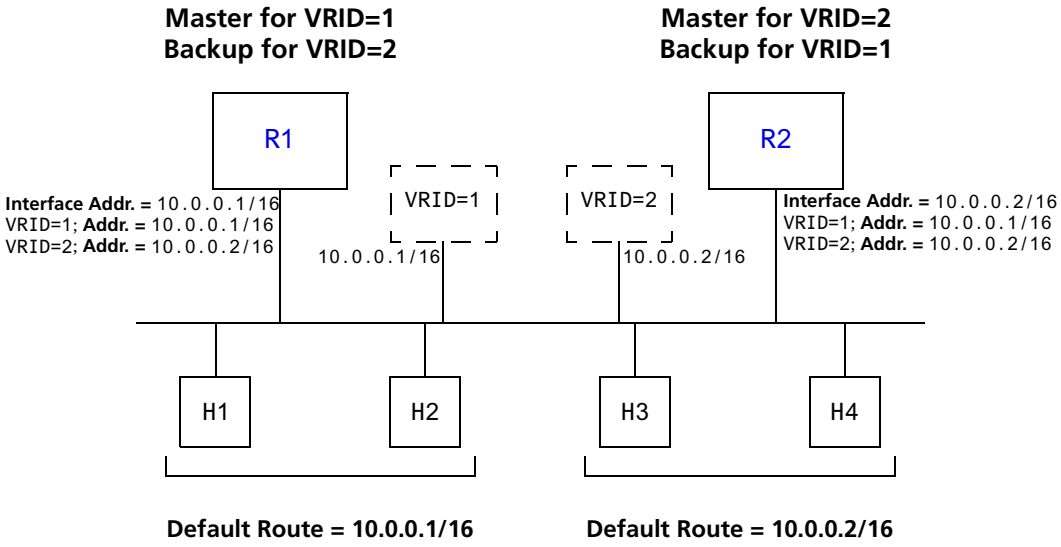


Figure 5. Symmetrical VRRP Configuration

In this configuration, half the hosts use 10.0.0.1/16 as their default route, and half use 10.0.0.2/16. IP address 10.0.0.1/16 is associated with virtual router `VRID=1`, and IP address 10.0.0.2/16 is associated with virtual router `VRID=2`.

If Router R1, the Master for virtual router `VRID=1`, goes down, Router R2 would take over the IP address 10.0.0.1/16. Similarly, if Router R2, the Master for virtual router `VRID=2`, goes down, Router R1 would take over the IP address 10.0.0.2/16.

Configuration of Router R1

The following is the configuration file for Router R1 in Figure 5.

```
1: interface create ip test address-netmask 10.0.0.1/16 port et.1.1
     !
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy create vrrp 2 interface test
     !
4: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
5: ip-redundancy associate vrrp 2 interface test address 10.0.0.2/16
     !
6: ip-redundancy start vrrp 1 interface test
7: ip-redundancy start vrrp 2 interface test
```

Router R1 is the owner of IP address 10.0.0.1/16. Line 4 associates this IP address with virtual router `VRID=1`, so Router R1 is the Master for virtual router `VRID=1`.

On line 5, Router R1 associates IP address 10.0.0.2/16 with virtual router `VRID=2`. However, since Router R1 does not own IP address 10.0.0.2/16, it is not the default Master for virtual router `VRID=2`.

Configuration of Router R2

The following is the configuration file for Router R2 in Figure 5.

```
1: interface create ip test address-netmask 10.0.0.2/16 port et.1.1
     !
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy create vrrp 2 interface test
     !
4: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
5: ip-redundancy associate vrrp 2 interface test address 10.0.0.2/16
     !
6: ip-redundancy start vrrp 1 interface test
7: ip-redundancy start vrrp 2 interface test
```

On line 1, Router R2 is made owner of IP address 10.0.0.2/16. Line 5 associates this IP address with virtual router `VRID=2`, so Router R2 is the Master for virtual router `VRID=2`. Line 4 associates IP address 10.0.0.1/16 with virtual router `VRID=1`, making Router R2 the Backup for virtual router `VRID=1`.

Multi-Backup Configuration

Figure 6 shows a VRRP configuration with three routers and three virtual routers. Each router serves as a Master for one virtual router and as a Backup for each of the others. When a Master router goes down, one of the Backups takes over the IP addresses of its virtual router.

In a VRRP configuration where more than one router is backing up a Master, you can specify which Backup router takes over when the Master goes down by setting the priority for the Backup routers.

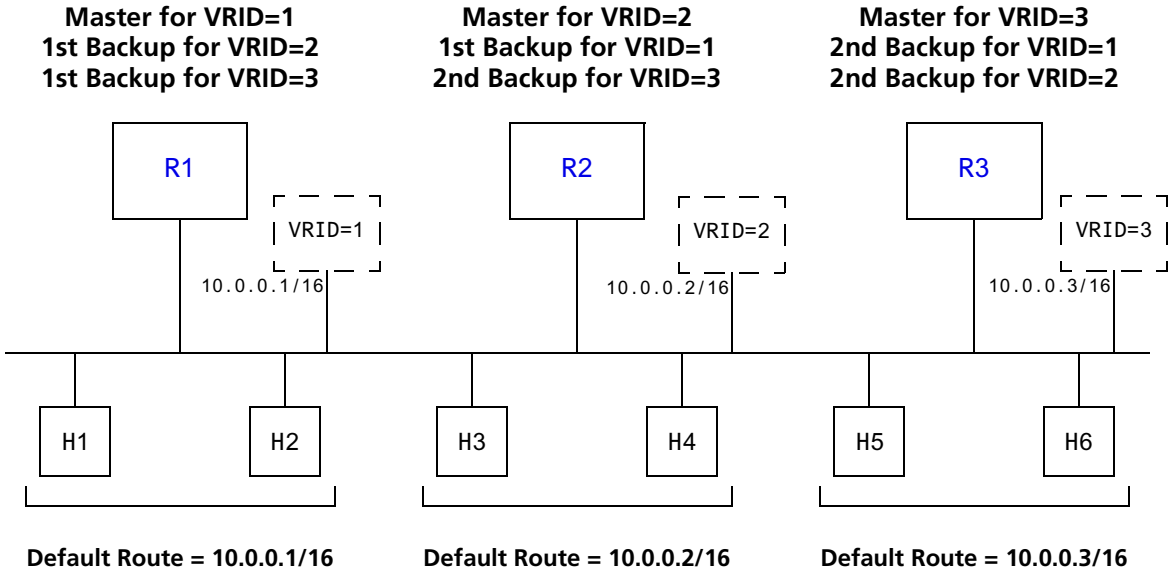


Figure 6. Multi-Backup VRRP Configuration

In this configuration, Router R1 is the Master for virtual router VRID=1 and the primary Backup for virtual routers VRID=2 and VRID=3. If Router R2 or R3 were to go down, Router R1 would assume the IP addresses associated with virtual routers VRID=2 and VRID=3.

Router R2 is the Master for virtual router VRID=2, the primary backup for virtual router VRID=1, and the secondary Backup for virtual router VRID=3. If Router R1 should fail, Router R2 would become the Master for virtual router VRID=1. If both Routers R1 and R3 should fail, Router R2 would become the Master for all three virtual routers. Packets sent to IP addresses 10.0.0.1/16, 10.0.0.2/16, and 10.0.0.3/16 would all go to Router R2.

Router R3 is the secondary Backup for virtual routers VRID=1 and VRID=2. It would become a Master router only if both Routers R1 and R2 should fail. In such a case, Router R3 would become the Master for all three virtual routers.

Configuration of Router R1

The following is the configuration file for Router R1 in Figure 6.

```
1: interface create ip test address-netmask 10.0.0.1/16 port et.1.1
     !
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy create vrrp 2 interface test
4: ip-redundancy create vrrp 3 interface test
     !
5: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
6: ip-redundancy associate vrrp 2 interface test address 10.0.0.2/16
7: ip-redundancy associate vrrp 3 interface test address 10.0.0.3/16
     !
8: ip-redundancy set vrrp 2 interface test priority 200
9: ip-redundancy set vrrp 3 interface test priority 200
     !
10: ip-redundancy start vrrp 1 interface test
11: ip-redundancy start vrrp 2 interface test
12: ip-redundancy start vrrp 3 interface test
```

Router R1's IP address on interface test is 10.0.0.1. There are three virtual routers on this interface:

- VRID=1 – IP address=10.0.0.1/16
- VRID=2 – IP address=10.0.0.2/16
- VRID=3 – IP address=10.0.0.3/16

Since the IP address of virtual router VRID=1 is the same as the interface's IP address (10.0.0.1), then the router automatically becomes the address owner of virtual router VRID=1.

A priority is associated with each of the virtual routers. The priority determines whether the router will become the Master or the Backup for a particular virtual router. Priorities can have values between 1 and 255. When a Master router goes down, the router with the next-highest priority takes over the virtual router. If more than one router has the next-highest priority, the router that has the highest-numbered interface IP address becomes the Master.

If a router is the address owner for a virtual router, then its priority for that virtual router is 255 and cannot be changed. If a router is ***not*** the address-owner for a virtual-router, then its priority for that virtual router is 100 by default, and can be changed by the user.

Since Router R1 is the owner of the IP address associated with virtual router VRID=1, it has a priority of 255 (the highest) for virtual router VRID=1. Lines 8 and 9 set Router R1's priority for virtual routers VRID=2 and VRID=3 at 200. If no other routers in the VRRP configuration have a higher priority, Router R1 will take over as Master for virtual routers VRID=2 and VRID=3, should Router R2 or R3 go down.

The following table shows the priorities for each virtual router configured on Router R1.

Virtual Router	Default Priority	Configured Priority
VRID=1 – IP address=10.0.0.1/16	255 (address owner)	255 (address owner)
VRID=2 – IP address=10.0.0.2/16	100	200 (see line 8)
VRID=3 – IP address=10.0.0.3/16	100	200 (see line 9)

Configuration of Router R2

The following is the configuration file for Router R2 in Figure 6.

```
1: interface create ip test address-netmask 10.0.0.2/16 port et.1.1
     !
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy create vrrp 2 interface test
4: ip-redundancy create vrrp 3 interface test
     !
5: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
6: ip-redundancy associate vrrp 2 interface test address 10.0.0.2/16
7: ip-redundancy associate vrrp 3 interface test address 10.0.0.3/16
     !
8: ip-redundancy set vrrp 1 interface test priority 200
9: ip-redundancy set vrrp 3 interface test priority 100
     !
10: ip-redundancy start vrrp 1 interface test
11: ip-redundancy start vrrp 2 interface test
12: ip-redundancy start vrrp 3 interface test
```

Line 8 sets the backup priority for virtual router VRID=1 to 200. Since this number is higher than Router R3's backup priority for virtual router VRID=1, Router R2 is the primary Backup, and Router R3 is the secondary Backup for virtual router VRID=1.

On line 9, the backup priority for virtual router VRID=3 is set to 100. Since Router R1's backup priority for this virtual router is 200, Router R1 is the primary Backup, and Router R2 is the secondary Backup for virtual router VRID=3.

The following table shows the priorities for each virtual router configured on Router R2.

Virtual Router	Default Priority	Configured Priority
VRID=1 – IP address=10.0.0.1/16	100	200 (see line 8)
VRID=2 – IP address=10.0.0.2/16	255 (address owner)	255 (address owner)
VRID=3 – IP address=10.0.0.3/16	100	100 (see line 9)

Note: Since 100 is the default priority, line 9, which sets the priority to 100, is actually unnecessary. It is included for illustration purposes only.

Configuration of Router R3

The following is the configuration file for Router R3 in Figure 6.

```
1: interface create ip test address-netmask 10.0.0.3/16 port et.1.1
     !
2: ip-redundancy create vrrp 1 interface test
3: ip-redundancy create vrrp 2 interface test
4: ip-redundancy create vrrp 3 interface test
     !
5: ip-redundancy associate vrrp 1 interface test address 10.0.0.1/16
6: ip-redundancy associate vrrp 2 interface test address 10.0.0.2/16
7: ip-redundancy associate vrrp 3 interface test address 10.0.0.3/16
     !
8: ip-redundancy set vrrp 1 interface test priority 100
9: ip-redundancy set vrrp 2 interface test priority 100
     !
10: ip-redundancy start vrrp 1 interface test
11: ip-redundancy start vrrp 2 interface test
12: ip-redundancy start vrrp 3 interface test
```

Lines 8 and 9 set the backup priority for Router R3 at 100 for virtual routers VRID=1 and VRID=2. Since Router R1 has a priority of 200 for backing up virtual router VRID=2, and Router R2 has a priority of 200 for backing up virtual router VRID=1, Router R3 is the secondary Backup for both virtual routers VRID=1 and VRID=2.

The following table shows the priorities for each virtual router configured on Router R3.

Virtual Router	Default Priority	Configured Priority
VRID=1 – IP address=10.0.0.1/16	100	100 (see line 8)
VRID=2 – IP address=10.0.0.2/16	100	100 (see line 9)
VRID=3 – IP address=10.0.0.3/16	255 (address owner)	255 (address owner)

Note: Since 100 is the default priority, lines 8 and 9, which set the priority to 100, are actually unnecessary. They are included for illustration purposes only.

Additional Configuration

This section covers settings you can modify in a VRRP configuration, including backup priority, advertisement interval, pre-empt mode, and authentication key.

Setting the Backup Priority

As described in "Multi-Backup Configuration" on page 74, you can specify which Backup router takes over when the Master router goes down by setting the priority for the Backup routers. To set the priority for a Backup router, enter the following command in Configure mode:

Set the Backup priority for a virtual router.	**ip-redundancy set vrrp** *<vrid>* **interface** *<interface>* **priority <number>**

The priority can be between 1 (lowest) and 254. The default is 100. The priority for the IP address owner is 255 and cannot be changed.

Setting the Advertisement Interval

The VRRP Master router sends periodic advertisement messages to let the other routers know that the Master is up and running. By default, advertisement messages are sent once each second. To change the VRRP advertisement interval, enter the following command in Configure mode:

Set the Advertisement interval for a virtual router.	**ip-redundancy set vrrp** *<vrid>* **interface** *<interface>* **adv-interval** *<seconds>*

Setting Pre-empt Mode

When a Master router goes down, the Backup with the highest priority takes over the IP addresses associated with the Master. By default, when the original Master comes back up again, it takes over from the Backup router that assumed its role as Master. When a VRRP router does this, it is said to be in *pre-empt mode*. Pre-empt mode is enabled by default on the GSR. You can prevent a VRRP router from taking over from a lower-priority Master by disabling pre-empt mode. To do this, enter the following command in Configure mode:

Disable pre-empt mode for a virtual router.	**ip-redundancy set vrrp** <*vrid*> **interface** <*interface*> **preempt-mode disabled**

Note: If the IP address owner is available, then it will always take over as the Master, regardless of whether pre-empt mode is on or off.

Setting an Authentication Key

By default, no authentication of VRRP packets is performed on the GSR. You can specify a clear-text password to be used to authenticate VRRP exchanges. To enable authentication, enter the following command in Configure mode:

Set an authentication key for a virtual router.	**ip-redundancy set vrrp** <*vrid*> **interface** <*interface*> **auth-type text auth-key** <*key*>

where <*key*> is a clear-text password.

Note: The GSR does not currently support the IP Authentication Header method of authentication.

Monitoring VRRP

The GSR provides two commands for monitoring a VRRP configuration: **ip-redundancy trace**, which displays messages when VRRP events occur, and **ip-redundancy show**, which reports statistics about virtual routers.

ip-redundancy trace

The **ip-redundancy trace** command is used for troubleshooting purposes. This command causes messages to be displayed when certain VRRP events occur on the GSR. To trace VRRP events, enter the following commands in Enable mode:

Display a message when any VRRP event occurs. (Disabled by default.)	**ip-redundancy trace vrrp events enabled**
Display a message when a VRRP router changes from one state to another; for example Backup to Master. (Enabled by default.)	**ip-redundancy trace vrrp state-transitions enabled**
Display a message when a VRRP packet error is detected. (Enabled by default.)	**ip-redundancy trace vrrp packet-errors enabled**
Enable all VRRP tracing.	**ip-redundancy trace vrrp all enabled**

ip-redundancy show

The **ip-redundancy show** command reports information about a VRRP configuration. To display VRRP information, enter the following commands in Enable mode.

Display information about all virtual routers.	**ip-redundancy show vrrp**
Display information about all virtual routers on a specified interface.	**ip-redundancy show vrrp interface** *<interface>*
Display detailed statistics about a specific virtual router	**ip-redundancy show vrrp** *<vrid>* **interface** *<interface>* **verbose**

VRRP Configuration Notes

- The Master router sends keep-alive advertisements. The frequency of these keep-alive advertisements is determined by setting the Advertisement interval parameter. The default value is 1 second.
- If a Backup router doesn't receive a keep-alive advertisement from the current Master within a certain period of time, it will transition to the Master state and start sending advertisements itself. The amount of time that a Backup router will wait before it becomes the new Master is based on the following equation:

  Master-down-interval = (3 * advertisement-interval) + skew-time

  The skew-time depends on the Backup router's configured priority:

  Skew-time = ((256 - Priority) / 256)

  Therefore, the higher the priority, the faster a Backup router will detect that the Master is down. For example:

  – Default advertisement-interval = 1 second

  – Default Backup router priority = 100

  – Master-down-interval = time it takes a Backup to detect the Master is down

    = (3 * adv-interval) + skew-time

    = (3 * 1 second) + ((256 - 100) / 256)

    = 3.6 seconds
- If a Master router is manually rebooted, or if its interface is manually brought down, it will send a special keep-alive advertisement that lets the Backup routers know that a new Master is needed immediately.
- A virtual router will respond to ARP requests with a virtual MAC address. This virtual MAC depends on the virtual router ID:

  virtual MAC address = 00005E:0001*XX*

  where *XX* is the virtual router ID

  This virtual MAC address is also used as the source MAC address of the keep-alive Advertisements transmitted by the Master router.
- If multiple virtual routers are created on a single interface, the virtual routers must have unique identifiers. If virtual routers are created on different interfaces, you can reuse virtual router IDs.

  For example, the following configuration is valid:

```
ip-redundancy create vrrp 1 interface test-A
ip-redundancy create vrrp 1 interface test-B
```

- As specified in RFC 2338, a Backup router that has transitioned to Master will not respond to pings, accept telnet sessions, or field SNMP requests directed at the virtual router's IP address.

  Not responding allows network management to notice that the original Master router (i.e., the IP address owner) is down.

Chapter 8

RIP Configuration Guide

RIP Overview

This chapter describes how to configure the Routing Information Protocol (RIP) on 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 GSR implements plain text and MD5 authentication methods for RIP Version 2.

The protocol independent features that apply to RIP are described in Chapter 6, "IP Routing Configuration Guide."

Configuring RIP

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

1. Start the RIP process by entering 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 GSR, 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 GSR 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>*
Enable automatic summarization and redistribution of RIP routes.	**rip set auto-summary disable|enable**
Specify broadcast of RIP packets regardless of number of interfaces present.	**rip set broadcast-state always|choose|never**
Check that reserved fields in incoming RIP V1 packets are zero.	**rip set check-zero disable|enable**
Enable acceptance of RIP routes that have a metric of zero.	**rip set check-zero-metric disable|enable**
Enable poison revers, as specified by RFC 1058.	**rip set poison-reverse disable|enable**

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.

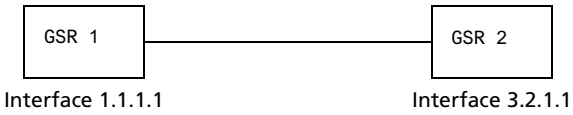
Monitoring RIP

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 9

OSPF Configuration Guide

OSPF Overview

Open Shortest Path First (OSPF) is a link-state routing protocol that supports IP subnetting and authentication. The GSR 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 GSR 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.

OSPF Multipath

The GSR also supports OSPF and static Multi-path. If multiple equal-cost OSPF or static routes have been defined for any destination, then the GSR "discovers" and uses all of them. The GSR will automatically learn up to four equal-cost OSPF or static routes and retain them in its forwarding information base (FIB). The forwarding module then installs flows for these destinations in a round-robin fashion.

Configuring OSPF

To configure OSPF on the GSR, 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:

- Enable OSPF.
- Create OSPF areas.
- Create an IP interface or assign an IP interface to a VLAN.
- Add IP interfaces to OSPF areas.
- 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 GSR has interfaces with differing bandwidths, the OSPF costs should be set accordingly.

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

Enabling OSPF

OSPF is disabled by default on the GSR.

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.

Table 4. OSPF Interface Parameters

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 GSR, 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 GSR 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 GSR lets you display 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 7 on page 101.

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 & 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 & Static Routes to OSPF

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

In the configuration shown in Figure 7 on page 101, 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
```

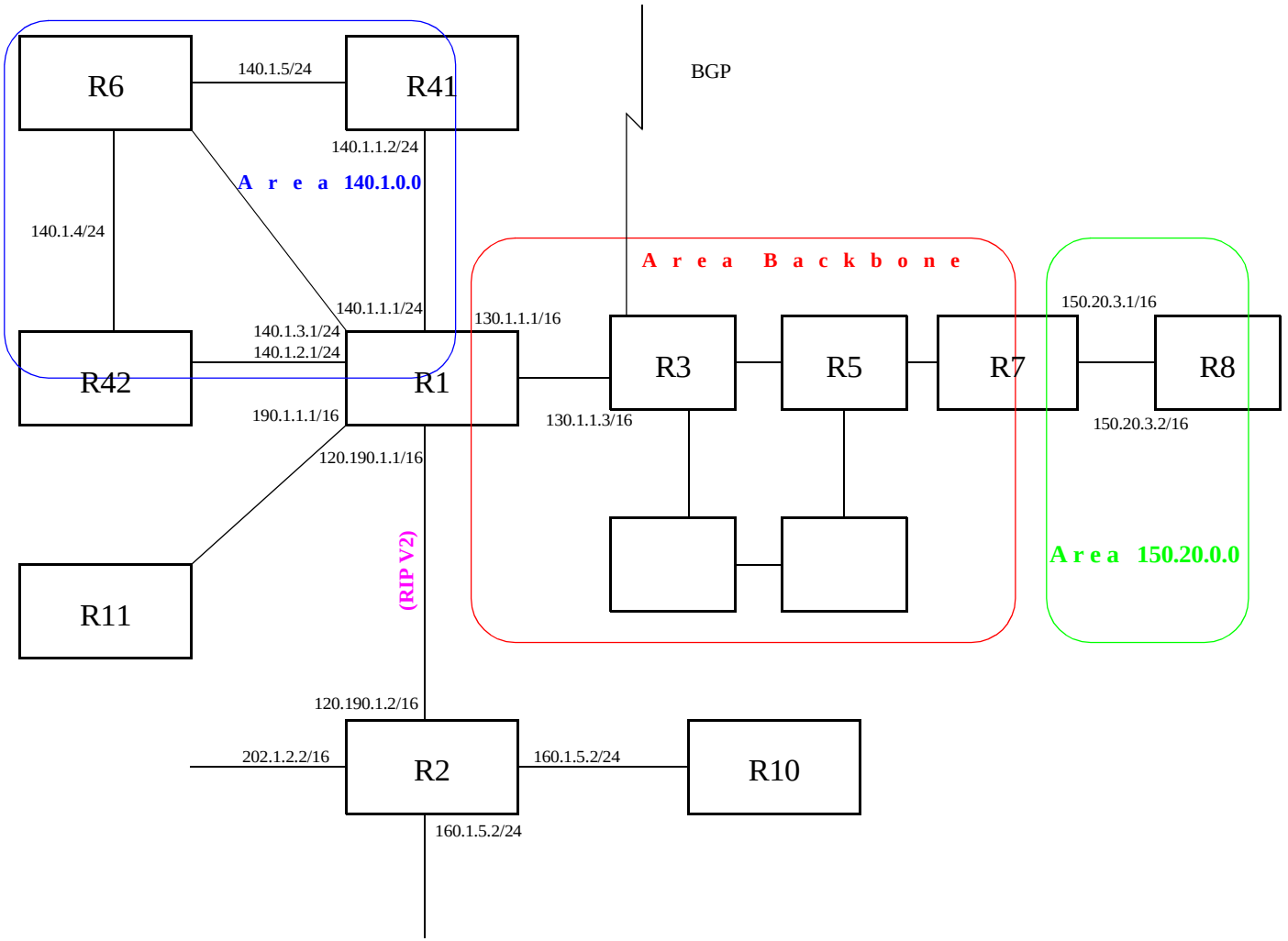
11. Create OSPF-ASE export source.

```
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 7. Exporting to OSPF



Chapter 10

BGP Configuration Guide

BGP Overview

The Border Gateway Protocol (BGP) is an exterior gateway protocol that allows IP routers to exchange network reachability information. BGP became an internet standard in 1989 (RFC 1105) and the current version, BGP-4, was published in 1994 (RFC 1771). BGP is typically run between Internet Service Providers. It is also frequently used by multi-homed ISP customers, as well as in large commercial networks.

Autonomous systems that wish to connect their networks together must agree on a method of exchanging routing information. Interior gateway protocols such as RIP and OSPF may be inadequate for this task since they were not designed to handle multi-AS, policy, and security issues. Similarly, using static routes may not be the best choice for exchanging AS-AS routing information because there may be a large number of routes, or the routes may change often.

Note: This chapter uses the term *Autonomous System* (AS) throughout. An AS is defined as a set of routers under a central technical administration that has a coherent interior routing plan and accurately portrays to other ASs what routing destinations are reachable by way of it.

In an environment where using static routes is not feasible, BGP is often the best choice for an AS-AS routing protocol. BGP prevents the introduction of routing loops created by multi-homed and meshed AS topologies. BGP also provides the ability to create and enforce policies at the AS level, such as selectively determining which AS routes are to be accepted or what routes are to be advertised to BGP peers.

The GSR BGP Implementation

The GSR routing protocol implementation is based on GateD 4.0.3 code (http://www.gated.org). GateD is a modular software program consisting of core services, a routing database, and protocol modules supporting multiple routing protocols (RIP versions 1 and 2, OSPF version 2, BGP version 2 through 4, and Integrated IS-IS).

Since the GSR IP routing code is based upon GateD, BGP can also be configured using a GateD configuration file (gated.conf) instead of the GSR Command Line Interface (CLI). Additionally, even if the GSR is configured using the CLI, the gated.conf equivalent can be displayed by entering the **ip-router show configuration-file** command at the GSR Enable prompt.

VLANs, interfaces, ACLs, and many other GSR configurable entities and functionality can only be configured using the GSR CLI. Therefore, a gated.conf file is dependent upon some GSR CLI configuration.

Basic BGP Tasks

This section describes the basic tasks necessary to configure BGP on the GSR. Due to the abstract nature of BGP, many BGP designs can be extremely complex. For any one BGP design challenge, there may only be one solution out of many that is relevant to common practice.

When designing a BGP configuration, it may be prudent to refer to information in RFCs, Internet drafts, and books about BGP. Some BGP designs may also require the aid of an experienced BGP network consultant.

Basic BGP configuration involves the following tasks:

- Setting the autonomous system number
- Setting the router ID
- Creating a BGP peer group
- Adding and removing a BGP peer host
- Starting BGP
- Using AS path regular expressions
- Using AS path prepend

Setting the Autonomous System Number

An autonomous system number identifies your autonomous system to other routers. To set the GSR's autonomous system number, enter the following command in Configure mode.

Set the GSR's autonomous system number.	**ip-router global set autonomous-system** <*num1*> **loops** <*num2*>

The a**utonomous-system** <*num1*> parameter sets the AS number for the router. Specify a number from 1–65534. The **loops** <*num2*> parameter controls the number of times the AS may appear in the as-path. The default is 1.

Setting the Router ID

The router ID uniquely identifies the GSR. To set the router ID to be used by BGP, enter the following command in Configure mode.

Set the GSR's router ID.	**ip-router global set router-id** <*hostname-or-IPaddr*>

If you do not explicitly specify the router ID, then an ID is chosen implicitly by the GSR. A secondary address on the loopback interface (the primary address being 127.0.0.1) is the most preferred candidate for selection as the GSR's router ID. If there are no secondary addresses on the loopback interface, then the default router ID is set to the address of the first interface that is in the up state that the GSR encounters (except the interface en0, which is the Control Module's interface). The address of a non point-to-point interface is preferred over the local address of a point-to-point interface. If the router ID is implicitly chosen to be the address of a non-loopback interface, and if that interface were to go down, then the router ID is changed. When the router ID changes, an OSPF router has to flush all its LSAs from the routing domain.

If you explicitly specify a router ID, then it would not change, even if all interfaces were to go down.

Configuring a BGP Peer Group

A BGP peer group is a group of neighbor routers that have the same update policies. To configure a BGP peer group, enter the following command in Configure mode:

Configure a BGP peer group.	**bgp create peer-group** <*number-or-string*> **type external | internal | igp | routing** **[autonomous-system** <*number*>**]** **[proto any | rip | ospf | static]** **[interface** <*interface-name-or-ipaddr*> **| all]**

where:

peer-group <*number-or-string*>
Is a group ID, which can be a number or a character string.

type Specifies the type of BGP group you are adding. You can specify one of the following:

external In the classic external BGP group, full policy checking is applied to all incoming and outgoing advertisements. The external neighbors must be directly reachable through one of the machine's local interfaces.

routing An internal group which uses the routes of an interior protocol to resolve forwarding addresses. Type Routing groups will determine the immediate next hops for routes by using the next hop received with a route from a peer as a forwarding address, and using this to look up an immediate next hop in an IGP's routes. Such groups support distant peers, but need to be informed of the IGP whose routes they are using to determine immediate next hops. This implementation comes closest to the IBGP implementation of other router vendors.

internal An internal group operating where there is no IP-level IGP, for example an SMDS network. Type Internal groups expect all peers to be directly attached to a shared subnet so that, like external peers, the next hops received in BGP advertisements may be used directly for forwarding. All Internal group peers should be L2 adjacent.

igp An internal group operating where there is no IP-level IGP; for example, an SMDS network.

autonomous-system <*number*>
Specifies the autonomous system of the peer group. Specify a number from 1 – 65534.

proto Specifies the interior protocol to be used to resolve BGP next hops. Specify one of the following:

- **any** Use any igp to resolve BGP next hops.
- **rip** Use RIP to resolve BGP next hops.
- **ospf** Use OSPF to resolve BGP next hops.
- **static** Use static to resolve BGP next hops.

interface <*name-or-IPaddr*> | **all**
Interfaces whose routes are carried via the IGP for which third-party next hops may be used instead. Use only for type Routing group. Specify the interface or **all** for all interfaces.

Adding and Removing a BGP Peer

There are two ways to add BGP peers to peer groups. You can explicitly add a peer host, or you can add a network. Adding a network allows for peer connections from any addresses in the range of network and mask pairs specified in the **bgp add network** command.

To add BGP peers to BGP peer groups, enter one of the following commands in Configure mode.

Add a host to a BGP peer group.	**bgp add peer-host** <*ipaddr*> **group** <*number-or-string*>
Add a network to a BGP peer group.	**bgp add network** <*ip-addr-mask*> **| all group** <*number-or-string*>

You may also remove a BGP peer from a peer group. To do so, enter the following command in Configure mode:

Remove a host from a BGP peer group.	**bgp clear peer-host <*ipaddr*>**

Starting BGP

BGP is disabled by default. To start BGP, enter the following command in Configure mode.

Start BGP.	**bgp start**

Using AS-Path Regular Expressions

An AS-path regular expression is a regular expression where the alphabet is the set of AS numbers. An AS-path regular expression is composed of one or more AS-path expressions. An AS-path expression is composed of AS path terms and AS-path operators.

An AS path term is one of the following three objects:

autonomous_system
: Is any valid autonomous system number, from one through 65534 inclusive.

. (dot)
: Matches any autonomous system number.

(aspath_regexp)
: Parentheses group subexpressions. An operator, such as * or ? works on a single element or on a regular expression enclosed in parentheses.

An AS-path operator is one of the following:

aspath_term {m,n}
: A regular expression followed by {m,n} (where m and n are both non-negative integers and m <= n) means at least m and at most n repetitions.

aspath_term {m}
: A regular expression followed by {m} (where m is a positive integer) means exactly m repetitions.

aspath_term {m,}
: A regular expression followed by {m,} (where m is a positive integer) means m or more repetitions.

aspath_term *
: An AS path term followed by * means zero or more repetitions. This is shorthand for {0,}.

aspath_term +
: A regular expression followed by + means one or more repetitions. This is shorthand for {1,}.

aspath_term ?
: A regular expression followed by ? means zero or one repetition. This is shorthand for {0,1}.

aspath_term | aspath_term
: Matches the AS term on the left, or the AS term on the right.

For example:

(4250 .*) Means anything beginning with 4250.

(.* 6301 .*) Means anything with 6301.

(.* 4250) Means anything ending with 4250.

(. * 1104 | 1125 | 1888 | 1135 .*)
Means anything containing 1104 or 1125 or 1888 or 1135.

AS-path regular expressions are used as one of the parameters for determining which routes are accepted and which routes are advertised.

AS-Path Regular Expression Examples

To import MCI routes with a preference of 165:

```
ip-router policy create bgp-import-source mciRoutes aspath-regular-
    expression "(.* 3561 .*)" origin any sequence-number 10
ip-router policy import source mciRoutes network all preference 165
```

To import all routes (.* matches all AS paths) with the default preference:

```
ip-router policy create bgp-import-source allOthers aspath-regular-
     expression "(.*)" origin any sequence-number 20
ip-router policy import source allOthers network all
```

To export all active routes from 284 or 813 or 814 or 815 or 816 or 3369 or 3561 to autonomous system 64800.

```
ip-router policy create bgp-export-destination to-64800 autonomous-
     system 64800
ip-router policy create aspath-export-source allRoutes aspath-regular-
     expression "(.*(284|813|814|815|816|3369|3561) .*)" origin any
     protocol all
ip-router policy export destination to-64800 source allRoutes network
     all
```

Using the AS Path Prepend Feature

When BGP compares two advertisements of the same prefix that have differing AS paths, the default action is to prefer the path with the lowest number of transit AS hops; in other words, the preference is for the shorter AS path length. The AS path prepend feature is a way to manipulate AS path attributes to influence downstream route selection. AS path prepend involves inserting the originating AS into the beginning of the AS prior to announcing the route to the exterior neighbor.

Lengthening the AS path makes the path less desirable than would otherwise be the case. However, this method of influencing downstream path selection is feasible only when comparing prefixes of the same length because an instance of a more specific prefix always is preferable.

On the GSR, the number of instances of an AS that are put in the route advertisement is controlled by the **as-count** option of the **bgp set peer-host** command.

The following is an example:

```
#
# insert two instances of the AS when advertising the route to this peer
#
bgp set peer-host 194.178.244.33 group nlnet as-count 2
#
# insert three instances of the AS when advertising the route to this
# peer
#
bgp set peer-host 194.109.86.5 group webnet as-count 3
```

Notes on Using the AS Path Prepend Feature

- Use the **as-count** option for external peer-hosts only.
- If the **as-count** option is entered for an active BGP session, routes will ***not*** be resent to reflect the new setting. To have routes reflect the new setting, you must restart the peer session. To do this:
  a. Enter Configure mode.
  b. Negate the command that adds the peer-host to the peer-group. (If this causes the number of peer-hosts in the peer-group to drop to zero, then you must also negate the command that creates the peer group.)
  c. Exit Configure mode.
  d. Re-enter Configure mode.

e. Add the peer-host back to the peer-group.

If the **as-count** option is part of the startup configuration, the above steps are unnecessary.

BGP Configuration Examples

This section presents sample configurations illustrating BGP features. The following features are demonstrated:

- BGP peering
- Internal BGP (IBGP)
- External BGP (EBGP) multihop
- BGP community attribute
- BGP local preference (local_pref) attribute
- BGP Multi-Exit Discriminator (MED) attribute
- EBGP aggregation
- Route reflection

BGP Peering Session Example

The router process used for a specific BGP peering session is known as a *BGP speaker*. A single router can have several BGP speakers. Successful BGP peering depends on the establishment of a neighbor relationship between BGP speakers. The first step in creating a BGP neighbor relationship is the establishment of a TCP connection (using TCP port 179) between peers.

A BGP Open message can then be sent between peers across the TCP connection to establish various BGP variables (BGP Version, AS number (ASN), hold time, BGP identifier, and optional parameters). Upon successful completion of the BGP Open negotiations, BGP Update messages containing the BGP routing table can be sent between peers.

BGP does not require a periodic refresh of the entire BGP routing table between peers. Only incremental routing changes are exchanged. Therefore, each BGP speaker is required to retain the entire BGP routing table of their peer for the duration of the peer's connection.

BGP "keepalive" messages are sent between peers periodically to ensure that the peers stay connected. If one of the routers encounters a fatal error condition, a BGP notification message is sent to its BGP peer, and the TCP connection is closed.

Figure 8 illustrates a sample BGP peering session.

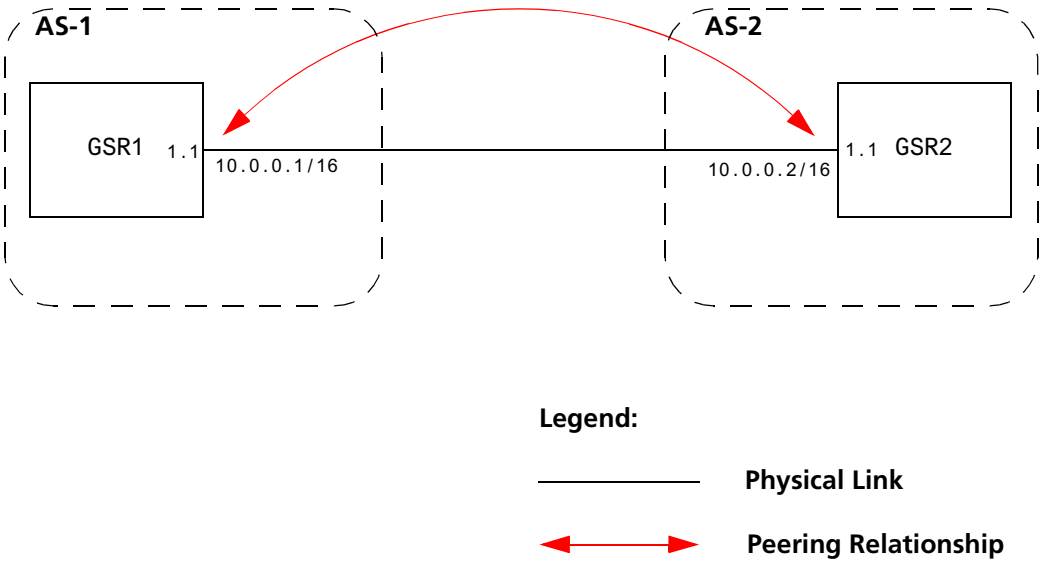


Figure 8. Sample BGP Peering Session

The CLI configuration for router GSR1 is as follows:

```
interface create ip et.1.1 address-netmask 10.0.0.1/16 port et.1.1
#
# Set the AS of the router
#
ip-router global set autonomous-system 1
#
# Set the router ID
#
ip-router global set router-id 10.0.0.1
#
# Create EBGP peer group pg1w2 for peering with AS 2
#
bgp create peer-group pg1w2 type external autonomous-system 2
#
# Add peer host 10.0.0.2 to group pg1w2
#
bgp add peer-host 10.0.0.2 group pg1w2
bgp start
```

The gated.conf file for router GSR1 is as follows:

```
autonomoussystem 1 ;
routerid 10.0.0.1 ;
bgp yes {
               group type external peeras 2
               {
                              peer 10.0.0.2
                                                ;
                };
};
```

The CLI configuration for router GSR2 is as follows:

```
interface create ip et.1.1 address-netmask 10.0.0.2/16 port et.1.1
ip-router global set autonomous-system 2
ip-router global set router-id 10.0.0.2
bgp create peer-group pg2w1 type external autonomous-system 1
bgp add peer-host 10.0.0.1 group pg2w1
bgp start
```

The gated.conf file for router GSR2 is as follows:

```
autonomoussystem 2 ;
routerid 10.0.0.2 ;
bgp yes {
               group type external peeras 1
               {
                              peer 10.0.0.1
                                                ;
                };
};
```

IBGP Configuration Example

Connections between BGP speakers within the same AS are referred to as internal links. A peer in the same AS is an internal peer. Internal BGP is commonly abbreviated IBGP; external BGP is EBGP.

An AS that has two or more EBGP peers is referred to as a multihomed AS. A multihomed AS can "transit" traffic between two ASs by advertising to one AS routes that it learned from the other AS. To successfully provide transit services, all EBGP speakers in the transit AS must have a consistent view of all of the routes reachable through their AS.

Multihomed transit ASs can use IBGP between EBGP-speaking routers in the AS to synchronize their routing tables. IBGP requires a full-mesh configuration; all EBGP speaking routers must have an IBGP peering session with every other EBGP speaking router in the AS.

An IGP, like OSPF, could possibly be used instead of IBGP to exchange routing information between EBGP speakers within an AS. However, injecting full Internet routes (50,000+ routes) into an IGP puts an expensive burden on the IGP routers. Additionally, IGPs cannot communicate all of the BGP attributes for a given route. It is, therefore, recommended that an IGP not be used to propagate full Internet routes between EBGP speakers. IBGP should be used instead.

IBGP Routing Group Example

An IBGP Routing group uses the routes of an interior protocol to resolve forwarding addresses. An IBGP Routing group will determine the immediate next hops for routes by using the next hop received with a route from a peer as a forwarding address, and using this to look up an immediate next hop in an IGP's routes. Such groups support distant peers, but need to be informed of the IGP whose routes they are using to determine immediate next hops. This implementation comes closest to the IBGP implementation of other router vendors.

You should use the IBGP Routing group as the mechanism to configure the GSR for IBGP. If the peers are directly connected, then IBGP using group-type Internal can also be used. Note that for running IBGP using group-type Routing you must run an IGP such as OSPF to resolve the next hops that come with external routes. You could also use protocol **any** so that all protocols are eligible to resolve the BGP forwarding address.

Figure 9 shows a sample BGP configuration that uses the Routing group type.

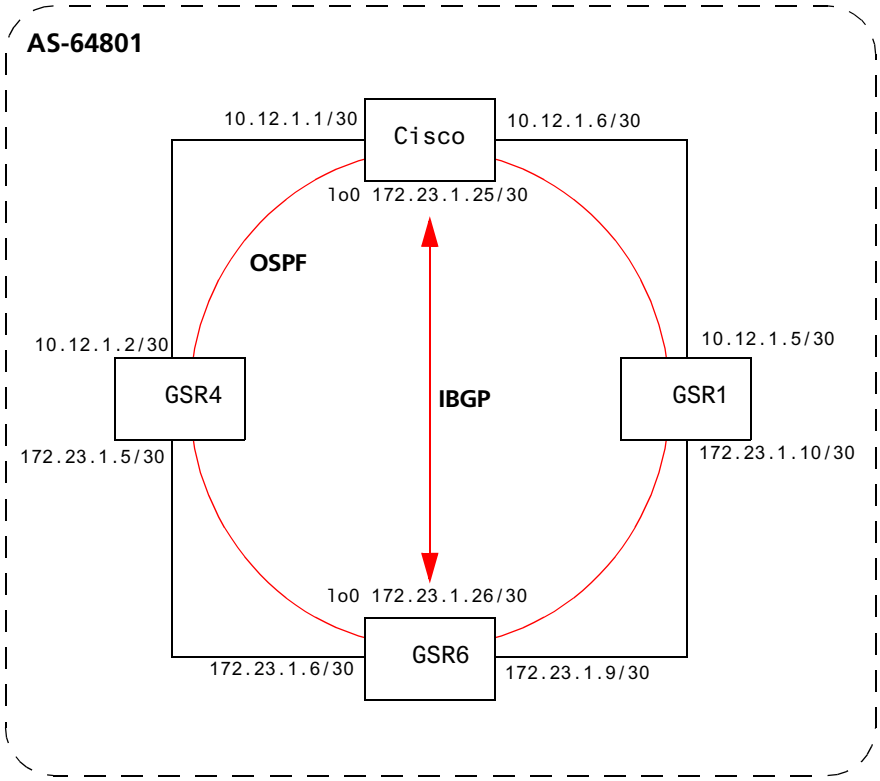


Figure 9. Sample IBGP Configuration (Routing Group Type)

In this example, OSPF is configured as the IGP in the autonomous system. The following lines in the router GSR6 configuration file configure OSPF:

```
#
# Create a secondary address for the loopback interface
#
interface add ip lo0 address-netmask 172.23.1.26/30
ospf create area backbone
ospf add interface to-GSR4 to-area backbone
ospf add interface to-GSR1 to-area backbone
#
# This line is necessary because we want CISCO to peer with our loopback
# address.This will make sure that the loopback address gets announced
# into OSPF domain
#
ospf add stub-host 172.23.1.26 to-area backbone cost 1
ospf set interface to-GSR4 priority 2
ospf set interface to-GSR1 priority 2
ospf set interface to-GSR4 cost 2
ospf start
```

The following lines in the Cisco router configure OSPF:

```
The following lines on the CISCO 4500 configures it for OSPF.
router ospf 1
 network 10.12.1.1 0.0.0.0 area 0
 network 10.12.1.6 0.0.0.0 area 0
 network 172.23.1.14 0.0.0.0 area 0
```

The following lines in the GSR6 set up peering with the Cisco router using the Routing group type.

```
# Create a internal routing group.
bgp create peer-group ibgp1 type routing autonomous-system 64801 proto any
interface all
# Add CISCO to the above group
bgp add peer-host 172.23.1.25 group ibgp1
# Set our local address. This line is necessary because we want CISCO to
# peer with our loopback
bgp set peer-group ibgp1 local-address 172.23.1.26
# Start BGP
bgp start
```

The following lines on the Cisco router set up IBGP peering with router GSR6.

```
router bgp 64801
!
! Disable synchronization between BGP and IGP
!
 no synchronization
neighbor 172.23.1.26 remote-as 64801
!
! Allow internal BGP sessions to use any operational interface for TCP
! connections
!
 neighbor 172.23.1.26 update-source Loopback0
```

IBGP Internal Group Example

The IBGP Internal group expects all peers to be directly attached to a shared subnet so that, like external peers, the next hops received in BGP advertisements may be used directly for forwarding. All Internal group peers should be L2 adjacent.

Figure 10 illustrates a sample IBGP Internal group configuration.

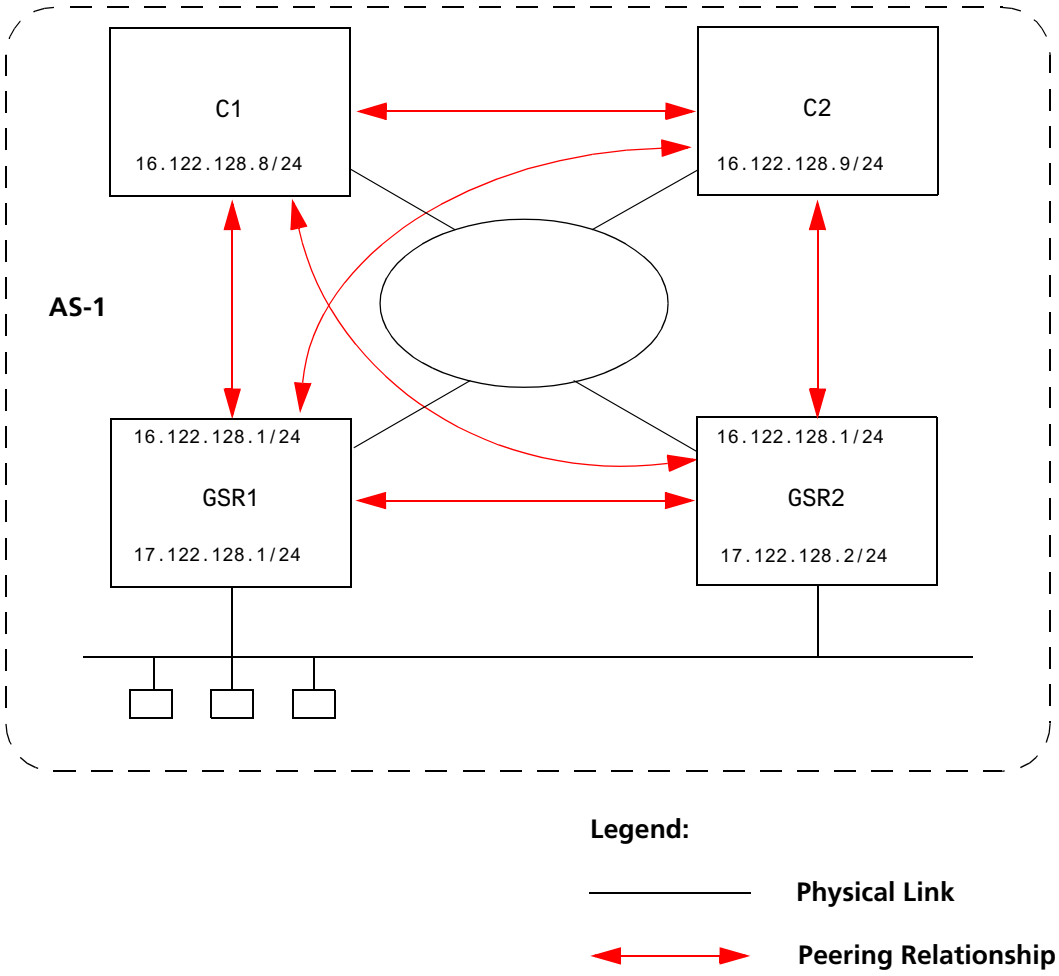


Figure 10. Sample IBGP Configuration (Internal Group Type)

The CLI configuration for router GSR1 is as follows:

```
ip-router global set autonomous-system 1
bgp create peer-group int-ibgp-1 type internal autonomous-system 1
bgp add peer-host 16.122.128.2 group int-ibgp-1
bgp add peer-host 16.122.128.8 group int-ibgp-1
bgp add peer-host 16.122.128.9 group int-ibgp-1
```

The gated.conf file for router GSR1 is as follows:

```
autonomoussystem 1 ;

routerid 16.122.128.1 ;

bgp yes {
         traceoptions aspath detail packets detail open detail update ;

         group type internal peeras 1
         {
                  peer 16.122.128.2
                           ;
                  peer 16.122.128.8
                           ;
                  peer 16.122.128.9
                           ;
         };
};
```

The CLI configuration for router GSR2 is as follows:

```
ip-router global set autonomous-system 1
bgp create peer-group int-ibgp-1 type internal autonomous-system 1
bgp add peer-host 16.122.128.1 group int-ibgp-1
bgp add peer-host 16.122.128.8 group int-ibgp-1
bgp add peer-host 16.122.128.9 group int-ibgp-1
```

The gated.conf file for router GSR2 is as follows:

```
autonomoussystem 1 ;

routerid 16.122.128.2 ;

bgp yes {
         traceoptions aspath detail packets detail open detail update ;

         group type internal peeras 1
         {
                  peer 16.122.128.1
                           ;
                  peer 16.122.128.8
                           ;
                  peer 16.122.128.9
                           ;
         };
};
```

The configuration for router C1 (a Cisco router) is as follows:

```
router bgp 1
 no synchronization
 network 16.122.128.0 mask 255.255.255.0
 network 17.122.128.0 mask 255.255.255.0
 neighbor 16.122.128.1 remote-as 1
 neighbor 16.122.128.1 next-hop-self
 neighbor 16.122.128.1 soft-reconfiguration inbound
 neighbor 16.122.128.2 remote-as 1
 neighbor 16.122.128.2 next-hop-self
 neighbor 16.122.128.2 soft-reconfiguration inbound
 neighbor 16.122.128.9 remote-as 1
 neighbor 16.122.128.9 next-hop-self
 neighbor 16.122.128.9 soft-reconfiguration inbound
 neighbor 18.122.128.4 remote-as 4
```

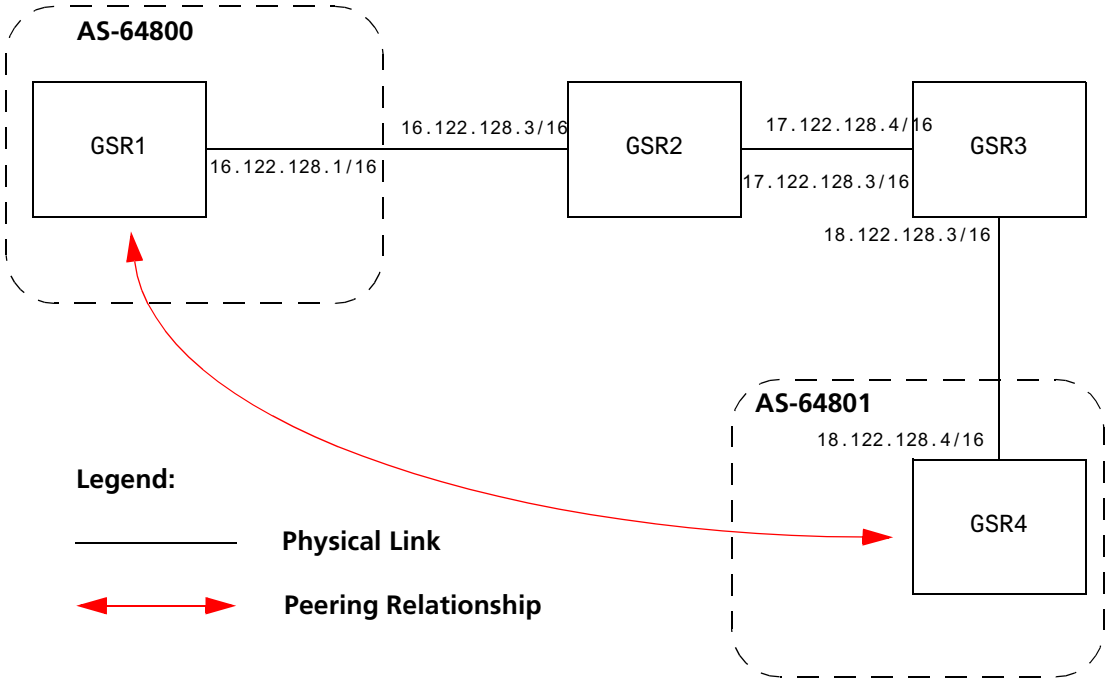
The configuration for router C2 (a Cisco router) is as follows:

```
router bgp 1
 no synchronization
 network 16.122.128.0 mask 255.255.255.0
 network 17.122.128.0 mask 255.255.255.0
 neighbor 14.122.128.5 remote-as 5
 neighbor 16.122.128.1 remote-as 1
 neighbor 16.122.128.1 next-hop-self
 neighbor 16.122.128.1 soft-reconfiguration inbound
 neighbor 16.122.128.2 remote-as 1
 neighbor 16.122.128.2 next-hop-self
 neighbor 16.122.128.2 soft-reconfiguration inbound
 neighbor 16.122.128.8 remote-as 1
 neighbor 16.122.128.8 next-hop-self
 neighbor 16.122.128.8 soft-reconfiguration inbound
```

EBGP Multihop Configuration Example

EBGP Multihop refers to a configuration where external BGP neighbors are not connected to the same subnet. Such neighbors are logically, but not physically connected. For example, BGP can be run between external neighbors across non-BGP routers. Some additional configuration is required to indicate that the external peers are not physically attached.

This sample configuration shows External BGP peers, GSR1 and GSR4, which are not connected to the same subnet.



The CLI configuration for router GSR1 is as follows:

```
bgp create peer-group ebgp_multihop autonomous-system 64801 type external
bgp add peer-host 18.122.128.2 group ebgp_multihop
!
! Specify the gateway option, which indicates EBGP multihop. Set the
! gateway option to the address of the router that has a route to the
! peer.
!
bgp set peer-host 18.122.128.2 gateway 16.122.128.3 group ebgp_multihop
```

The gated.conf file for router GSR1 is as follows:

```
autonomoussystem 64800 ;

routerid 0.0.0.1 ;

bgp yes {
         traceoptions state ;

         group type external peeras 64801
         {
                 peer 18.122.128.2
                         gateway 16.122.128.3
                         ;
         };
};

static {
         18.122.0.0 masklen 16
                 gateway 16.122.128.3
                 ;
};
```

The CLI configuration for router GSR2 is as follows:

```
interface create ip to-R1 address-netmask 16.122.128.3/16 port et.1.1
interface create ip to-R3 address-netmask 17.122.128.3/16 port et.1.2
#
# Static route needed to reach 18.122.0.0/16
#
ip add route 18.122.0.0/16 gateway 17.122.128.4
```

The gated.conf file for router GSR2 is as follows:

```
static {
        18.122.0.0 masklen 16
                 gateway 17.122.128.4
                 ;
};
```

The CLI configuration for router GSR3 is as follows:

```
interface create ip to-R2 address-netmask 17.122.128.4/16 port et.4.2
interface create ip to-R4 address-netmask 18.122.128.4/16 port et.4.4
ip add route 16.122.0.0/16 gateway 17.122.128.3
```

The gated.conf file for router GSR3 is as follows:

```
static {
         16.122.0.0 masklen 16
                 gateway 17.122.128.3
                 ;
};
```

The CLI configuration for router GSR4 is as follows:

```
bgp create peer-group ebgp_multihop autonomous-system 64801 type external
bgp add peer-host 18.122.128.2 group ebgp_multihop
!
! Specify the gateway option, which indicates EBGP multihop. Set the
! gateway option to the address of the router that has a route to the
! peer.
!
bgp set peer-host 18.122.128.2 gateway 16.122.128.3 group ebgp_multihop
```

The gated.conf file for router GSR4 is as follows:

```
autonomoussystem 64800 ;

routerid 0.0.0.1 ;

bgp yes {
          traceoptions state ;

          group type external peeras 64801
          {
                  peer 18.122.128.2
                          gateway 16.122.128.3
```

Community Attribute Example

The following configuration illustrates the BGP community attribute. Community is specified as one of the parameters in the **optional attributes list** option of the **ip-router policy create** command.

Figure 11 shows a BGP configuration where the specific community attribute is used. Figure 12 shows a BGP configuration where the well-known community attribute is used.

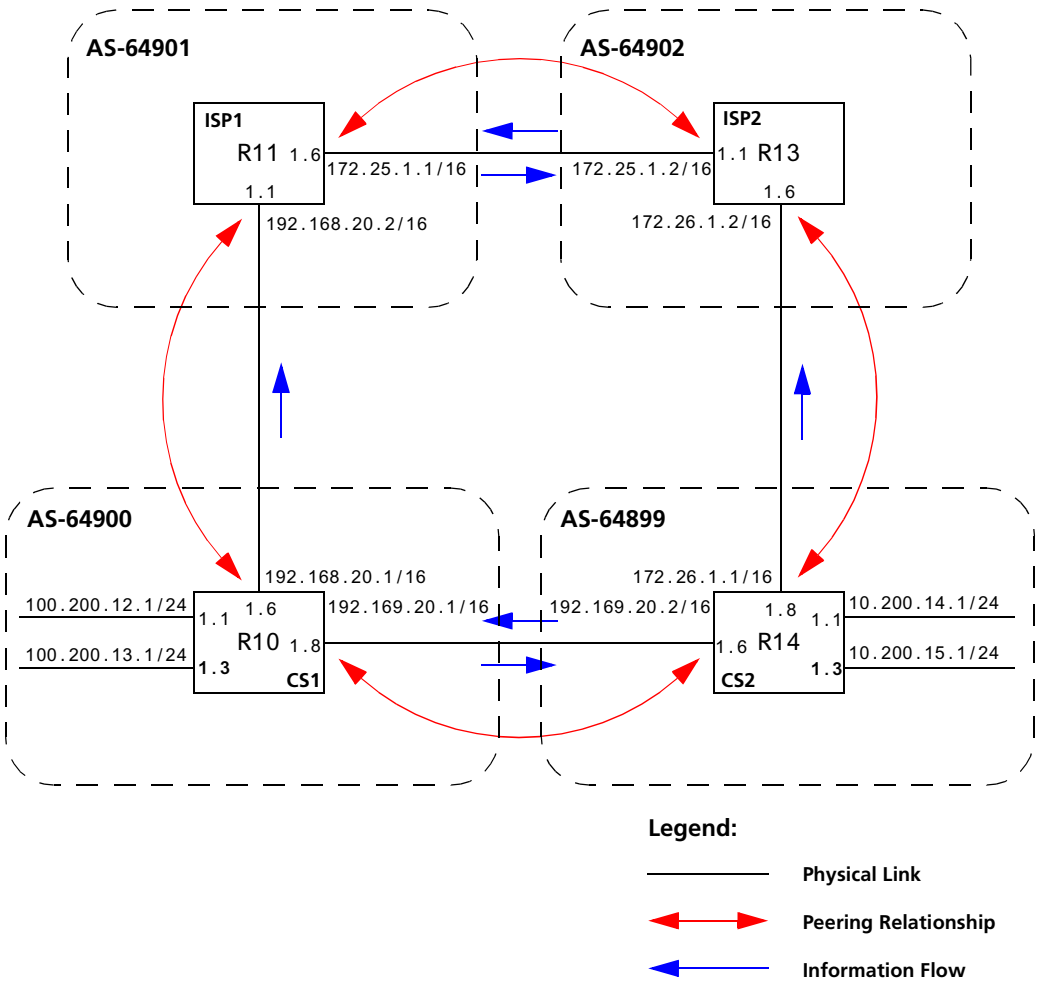


Figure 11. Sample BGP Configuration (Specific Community)

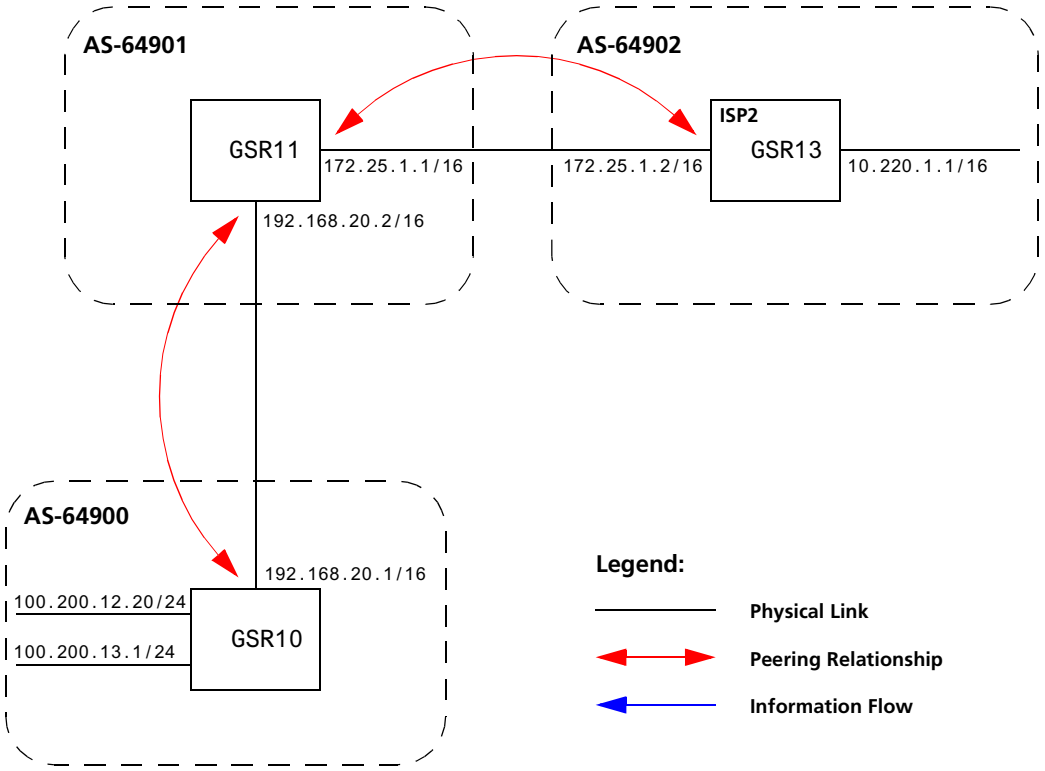


Figure 12. Sample BGP Configuration (Well-Known Community)

The Community attribute can be used in three ways:

1. In a BGP Group statement: Any packets sent to this group of BGP peers will have the communities attribute in the BGP packet modified to be this communities attribute value from this AS.

2. In an Import Statement: Any packets received from a BGP peer will be checked for the community attribute. The **optional-attributes-list** option of the **ip-router policy create** command allows the specification of an import policy based on optional path attributes (for instance, the community attribute) found in the BGP update. If multiple communities are specified in the **optional-attributes-list** option, only updates carrying all of the specified communities will be matched. If **well-known-community none** is specified, only updates lacking the community attribute will be matched.

   Note that it is quite possible for several BGP import clauses to match a given update. If more than one clause matches, the first matching clause will be used; all later matching clauses will be ignored. For this reason, it is generally desirable to order import clauses from most to least specific. An import clause without an **optional-attributes-list** option will match any update with any (or no) communities.

In Figure 12, router GSR11 has the following configuration:

```
#
# Create an optional attribute list with identifier color1 for a community
# attribute (community-id 160 AS 64901)
#
ip-router policy create optional-attributes-list color1 community-id 160
    autonomous-system 64901
#
# Create an optional attribute list with identifier color2 for a community
# attribute (community-id 155 AS 64901)
#
ip-router policy create optional-attributes-list color2 community-id 155
    autonomous-system 64901
#
# Create a BGP import source for importing routes from AS 64900 containing the
# community attribute (community-id 160 AS 64901). This import source is given an
# identifier 901color1 and sequence-number 1.
#
ip-router policy create bgp-import-source 901color1 optional-attributes-list
    color1 autonomous-system 64900 sequence-number 1
ip-router policy create bgp-import-source 901color2 optional-attributes-list
    color2 autonomous-system 64900 sequence-number 2
ip-router policy create bgp-import-source 901color3 optional-attributes-list
    color1 autonomous-system 64902 sequence-number 3
ip-router policy create bgp-import-source 901color4 optional-attributes-list
    color2 autonomous-system 64902 sequence-number 4
#
# Import all routes matching BGP import source 901color1 (from AS 64900 having
# community attribute with ID 160 AS 64901) with a preference of 160
#
ip-router policy import source 901color1 network all preference 160
ip-router policy import source 901color2 network all preference 155
ip-router policy import source 901color3 network all preference 160
ip-router policy import source 901color4 network all preference 155
```

In Figure 12, router GSR13 has the following configuration:

```
ip-router policy create optional-attributes-list color1 community-id 160
   autonomous-system 64902
ip-router policy create optional-attributes-list color2 community-id 155
   autonomous-system 64902
ip-router policy create bgp-import-source 902color1 optional-attributes-list
   color1 autonomous-system 64899 sequence-number 1
ip-router policy create bgp-import-source 902color2 optional-attributes-list
   color2 autonomous-system 64899 sequence-number 2
ip-router policy create bgp-import-source 902color3 optional-attributes-list
   color1 autonomous-system 64901 sequence-number 3
ip-router policy create bgp-import-source 902color4 optional-attributes-list
   color2 autonomous-system 64901 sequence-number 4
ip-router policy import source 902color1 network all preference 160
ip-router policy import source 902color2 network all preference 155
ip-router policy import source 902color3 network all preference 160
ip-router policy import source 902color4 network all preference 155
```

3. In an Export Statement: The **optional-attributes-list** option of the **ip-router policy create bgp-export-destination** command may be used to send the BGP community attribute. Any communities specified with the **optional-attributes-list** option are sent in addition to any received in the route or specified with the group.

In Figure 12, router GSR10 has the following configuration:

```
#
# Create an optional attribute list with identifier color1 for a community
# attribute (community-id 160 AS 64902)
#
ip-router policy create optional-attributes-list color1 community-id 160
    autonomous-system 64902
#
# Create an optional attribute list with identifier color2 for a community
# attribute (community-id 155 AS 64902)
#
ip-router policy create optional-attributes-list color2 community-id 155
    autonomous-system 64902
#
# Create a direct export source
#
ip-router policy create direct-export-source 900toanydir metric 10
#
# Create BGP export-destination for exporting routes to AS 64899 containing the
# community attribute (community-id 160 AS 64902). This export-destination has an
# identifier 900to899dest
#
ip-router policy create bgp-export-destination 900to899dest autonomous-system
    64899 optional-attributes-list color1
ip-router policy create bgp-export-destination 900to901dest autonomous-system
    64901 optional-attributes-list color2
#
# Export routes to AS 64899 with the community attribute (community-id 160 AS
# 64902)
#
ip-router policy export destination 900to899dest source 900toanydir network all
ip-router policy export destination 900to901dest source 900toanydir network all
```

In Figure 12, router GSR14 has the following configuration:

```
ip-router policy create bgp-export-destination 899to900dest autonomous-system
    64900 optional-attributes-list color1
ip-router policy create bgp-export-destination 899to902dest autonomous-system
    64902 optional-attributes-list color2
ip-router policy create bgp-export-source 900toany autonomous-system 64900 metric
    10
ip-router policy create optional-attributes-list color1 community-id 160
    autonomous-system 64901
ip-router policy create optional-attributes-list color2 community-id 155
    autonomous-system 64901
ip-router policy export destination 899to900dest source 899toanydir network all
ip-router policy export destination 899to902dest source 899toanydir network all
```

Any communities specified with the **optional-attributes-list** option are sent in addition to any received with the route or associated with a BGP export destination.

The community attribute may be a single community or a set of communities. A maximum of 10 communities may be specified.

The community attribute can take any of the following forms:

- Specific community

  The specific community consists of the combination of the AS-value and community ID.

- Well-known-community no-export

  Well-known-community no-export is a special community which indicates that the routes associated with this attribute must not be advertised outside a BGP confederation boundary. Since the GSR's implementation does not support Confederations, this boundary is an AS boundary.

  For example, router GSR10 in Figure 12 has the following configuration:

```
ip-router policy create optional-attributes-list noexport well-known-
    community no-export
ip-router policy create bgp-export-destination 900to901dest autonomous-
    system 64901 optional-attributes-list noexport
ip-router policy export destination 900to901dest source 900to901src
    network all
ip-router policy export destination 900to901dest source 900to901dir
    network all
```

- Well-known-community no-advertise

  Well-known-community no-advertise is a special community indicating that the routes associated with this attribute must not be advertised to other bgp peers. A packet can be modified to contain this attribute and passed to its neighbor. However, if a packet is received with this attribute, it cannot be transmitted to another BGP peer.

- Well-known-community no-export-subconfed

  Well-known-community no-export-subconfed is a special community indicating the routes associated with this attribute must not be advertised to external BGP peers. (This includes peers in other members' autonomous systems inside a BGP confederation.)

  A packet can be modified to contain this attribute and passed to its neighbor. However, if a packet is received with this attribute, the routes (prefix-attribute pair) cannot be advertised to an external BGP peer.

- Well-known-community none

  This is not actually a community, but rather a keyword that specifies that a received BGP update is only to be matched if no communities are present. It has no effect when originating communities.

Notes on Using Communities

When originating BGP communities, the set of communities that is actually sent is the union of the communities received with the route (if any), those specified in group policy (if any), and those specified in export policy (if any).

When receiving BGP communities, the update is only matched if all communities specified in the **optional-attributes-list** option of the **ip-router policy create** command are present in the BGP update. (If additional communities are also present in the update, it will still be matched.)

Local_Pref Attribute Example

Figure 13 shows a BGP configuration that uses the BGP local preference (Local_Pref) attribute in a sample BGP configuration with two autonomous systems.

The local preference is not set directly in the CLI, but rather is a function of the GateD preference and setpref metric. The setpref option allows GateD to set the local preference to reflect GateD's own internal preference for the route, as given by the global protocol preference value. The setpref option may be used with routing or internal type groups. BGP routes with a larger Local_Pref are preferred.

The formula used to compute the local preference is as follows:

Local_Pref = 254 – (global protocol preference for this route) + set preference metric

Note: A value greater than 254 will be reset to 254. GateD will only send Local_Pref values between 0 and 254.

In a mixed GateD and non-GateD network, the non-GateD IBGP implementation may send Local_Pref values that are greater than 254. When operating a mixed network of this type, you should make sure that all routers are restricted to sending Local_Pref values in the range metric to 254.

In the sample network in Figure 13, all the traffic exits Autonomous System 64901 through the link between router GSR13 and router GSR11. This is accomplished by setting the Local_Pref attribute.

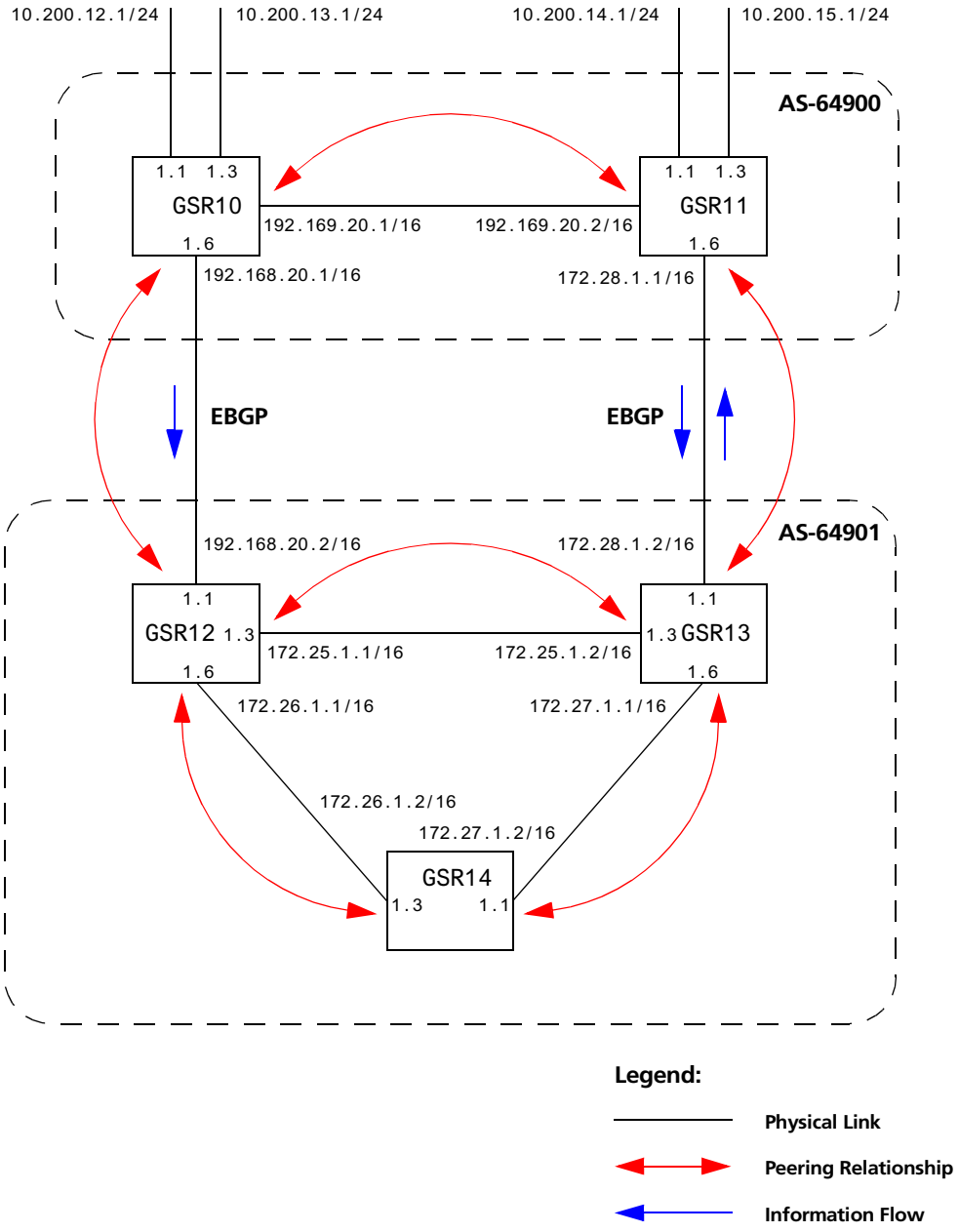


Figure 13. Sample BGP Configuration (Local_Pref Attribute)

In router GSR12's CLI configuration file, the import preference is set to 160:

```
#
# Set the set-pref metric for the IBGP peer group
#
bgp set peer-group as901 set-pref 100
ip-router policy create bgp-import-source as900 autonomous-system 64900
    preference 160
```

Using the formula for local preference [Local_Pref = 254 - (global protocol preference for this route) + metric], the Local_Pref value put out by router GSR12 is 254 - 160+100 = 194.

For router GSR13, the import preference is set to 150. The Local_Pref value put out by router GSR12 is 254 - 160+100 = 204.

```
ip-router policy create bgp-import-source as900 autonomous-system 64900
    preference 150
```

Notes on Using the Local_Pref Attribute

- All routers in the same network that are running GateD and participating in IBGP should use the setpref metric, and the setpref metric should be set to the same value.

  For example, in Figure 13, routers GSR12, GSR13, and GSR14 have the following line in their CLI configuration files:

  ```
  bgp set peer-group as901 set-pref 100
  ```

- The value of the setpref metric should be consistent with the import policy in the network.

  The metric value should be set high enough to avoid conflicts between BGP routes and IGP or static routes. For example, if the import policy sets GateD preferences ranging from 170 to 200, a setpref metric of 170 would make sense. You should set the metric high enough to avoid conflicts between BGP routes and IGP or static routes.

Multi-Exit Discriminator Attribute Example

Multi-Exit Discriminator (MED) is a BGP attribute that affects the route selection process. MED is used on external links to discriminate among multiple exit or entry points to the same neighboring AS. All other factors being equal, the exit or entry point with a lower metric should be preferred. If received over external links, the MED attribute may be propagated over internal links to other BGP speakers within the same AS. The MED attribute is never propagated to other BGP speakers in neighboring autonomous systems.

Figure 14 shows a sample BGP configuration where the MED attribute has been used.

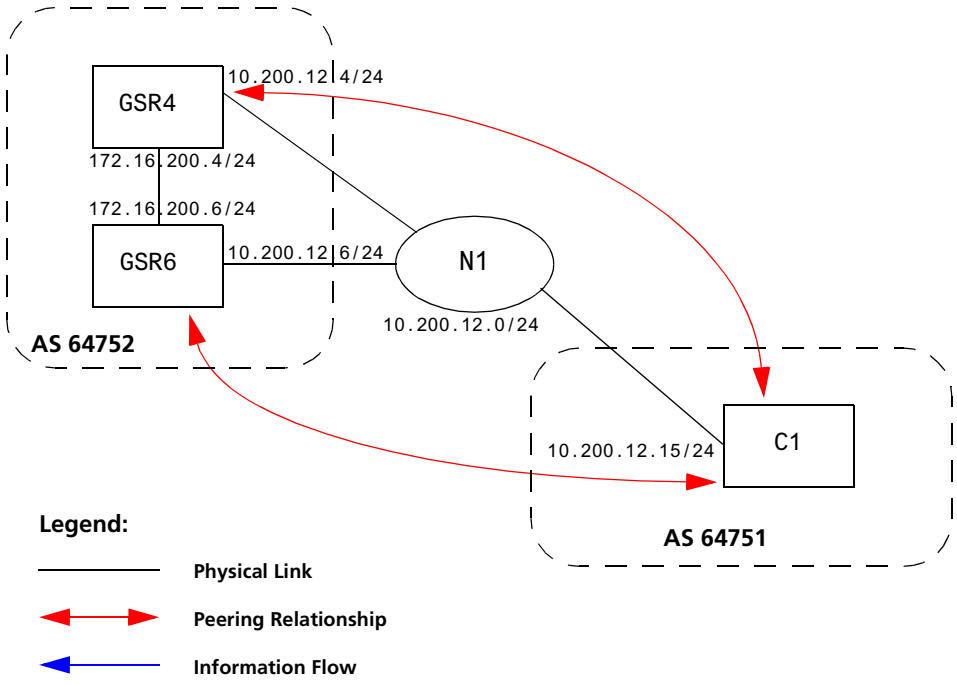


Figure 14. Sample BGP Configuration (MED Attribute)

Routers GSR4 and GSR6 inform router C1 about network 172.16.200.0/24 through External BGP (EBGP). Router GSR6 announced the route with a MED of 10, whereas router GSR4 announces the route with a MED of 20. Of the two EBGP routes, router C1 chooses the one with a smaller MED. Thus router C1 prefers the route from router GSR6, which has a MED of 10.

Router GSR4 has the following CLI configuration:

```
bgp create peer-group pg752to751 type external autonomous-system 64751
bgp add peer-host 10.200.12.15 group pg752to751
#
# Set the MED to be announced to peer group pg752to751
#
bgp set peer-group pg752to751 metric-out 20
```

Router GSR6 has the following CLI configuration:

```
bgp create peer-group pg752to751 type external autonomous-system 64751
bgp add peer-host 10.200.12.15 group pg752to751
bgp set peer-group pg752to751 metric-out 10
```

EBGP Aggregation Example

Figure 15 shows a simple EBGP configuration in which one peer is exporting an aggregated route to its upstream peer and restricting the advertisement of contributing routes to the same peer. The aggregated route is 212.19.192.0/19.

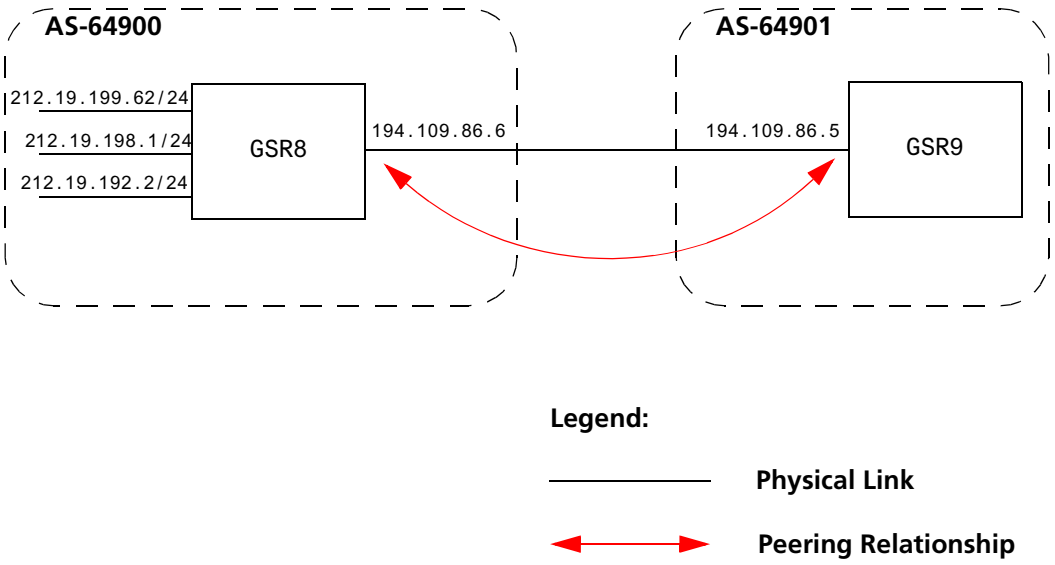


Figure 15. Sample BGP Configuration (Route Aggregation)

Router GSR8 has the following CLI configuration:

```
interface add ip xleapnl address-netmask 212.19.192.2/24
interface create ip hobbygate address-netmask 212.19.199.62/24 port
    et.1.2
interface create ip xenosite address-netmask 212.19.198.1/24 port
    et.1.7
interface add ip lo0 address-netmask 212.19.192.1/30
bgp create peer-group webnet type external autonomous system 64901
bgp add peer-host 194.109.86.5 group webnet
#
# Create an aggregate route for 212.19.192.0/19 with all its subnets as
# contributing routes
#
ip-router policy summarize route 212.19.192.0/19
ip-router policy redistribute from-proto aggregate to-proto bgp target-
    as 64901 network 212.19.192.0/19
ip-router policy redistribute from-proto direct to-proto bgp target-as
    64901 network all restrict
```

Router GSR9 has the following CLI configuration:

```
bgp create peer-group rtr8 type external autonomous system 64900
bgp add peer-host 194.109.86.6 group rtr8
```

Route Reflection Example

In some ISP networks, the internal BGP mesh becomes quite large, and the IBGP full mesh does not scale well. For such situations, route reflection provides a way to alleviate the need for a full IBGP mesh. In route reflection, the clients peer with the route reflector and exchange routing information with it. In turn, the route reflector passes on (reflects) information between clients.

The IBGP peers of the route reflector fall under two categories: clients and non-clients. A route reflector and its clients form a cluster. All peers of the route reflector that are not part of the cluster are non-clients. The GSR supports client peers as well as non-client peers of a route reflector.

Figure 16 shows a sample configuration that uses route reflection.

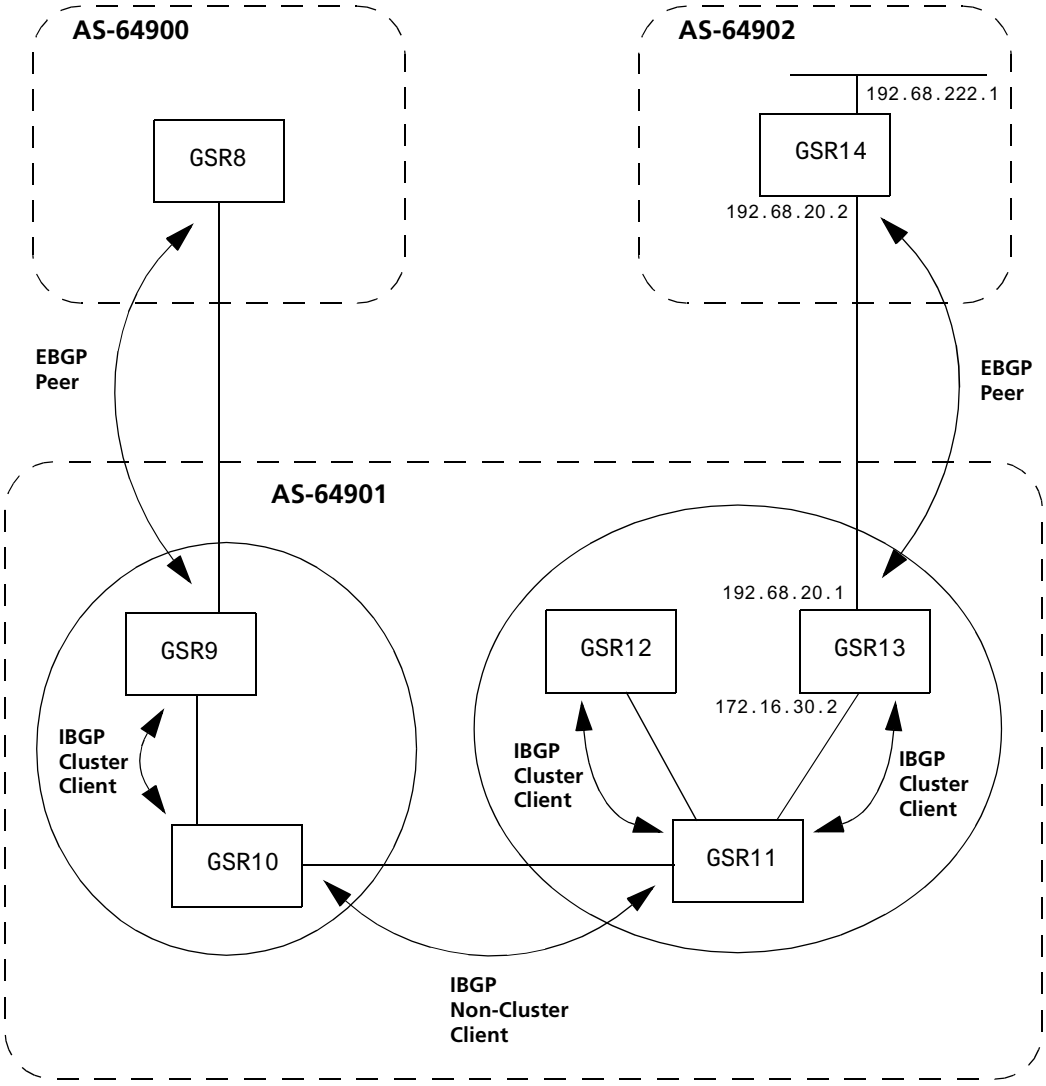


Figure 16. Sample BGP Configuration (Route Reflection)

In this example, there are two clusters. Router GSR10 is the route reflector for the first cluster and router GSR11 is the route reflector for the second cluster. Router GSR10 has router GSR9 as a client peer and router GSR11 as a non-client peer.

The following line in router GSR10's configuration file causes it to be a route reflector.

```
bgp set peer-group GSR9 reflector-client
```

Router GSR11 has router GSR12 and router GSR13 as client peers and router GSR10 as non-client peer. The following line in router GSR11's configuration file specifies it to be a route reflector

```
bgp set peer-group rtr11 reflector-client
```

Even though the IBGP Peers are not fully meshed in AS 64901, the direct routes of router GSR14, that is, 192.68.222.0/24 in AS 64902 (which are redistributed in BGP) do show up in the route table of router GSR8 in AS64900, as shown below:

```
*****************************************
* Route Table (FIB) of Router 8         *
*****************************************
rtr-8# ip show routes

Destination               Gateway                   Owner     Netif
-----------               -------                   -----     -----
10.50.0.0/16              directly connected        -         en
127.0.0.0/8               127.0.0.1                 Static    lo
127.0.0.1                 127.0.0.1                 -         lo
172.16.20.0/24            directly connected        -         mls1
172.16.70.0/24            172.16.20.2               BGP       mls1
172.16.220.0/24           172.16.20.2               BGP       mls1
192.68.11.0/24            directly connected        -         mls0
192.68.20.0/24            172.16.20.2               BGP       mls1
192.68.222.0/24           172.16.20.2               BGP       mls1
```

The direct routes of router GSR8, i.e. 192.68.11.0/24 in AS64900 (which are redistributed in BGP), do show up in the route table of router GSR14 in AS64902, as shown below:

```
*********************************************************
*  Route Table (FIB) of Router 14                       *
*********************************************************
rtr-14# ip show routes

Destination          Gateway              Owner     Netif
-----------          -------              -----     -----
10.50.0.0/16         directly connected   -         en0
127.0.0.0/8          127.0.0.1            Static    lo0
127.0.0.1            127.0.0.1            -         lo0
172.16.20.0/24       192.68.20.1          BGP       mls1
172.16.30.0/24       192.68.20.1          BGP       mls1
172.16.90.0/24       192.68.20.1          BGP       mls1
192.68.11.0/24       192.68.20.1          BGP       mls1
192.68.20.0/24       directly connected   -         mls1
192.68.222.0/24      directly connected   -         mls0
```

Notes on Using Route Reflection

- Two types of route reflection are supported:
  - By default, all routes received by the route reflector from a client are sent to all internal peers (including the client's group, but not the client itself).
  - If the **no-client-reflec**t option is enabled, routes received from a route reflection client are sent only to internal peers that are not members of the client's group. In this case, the client's group must itself be fully meshed.

  In either case, all routes received from a non-client internal peer are sent to all route reflection clients.
- Typically, a single router acts as the reflector for a cluster of clients. However, for redundancy, two or more may also be configured to be reflectors for the same cluster. In this case, a cluster ID should be selected to identify all reflectors serving the cluster, using the **clusterid** option. Gratuitous use of multiple redundant reflectors is not advised, since it can lead to an increase in the memory required to store routes on the redundant reflectors' peers.
- No special configuration is required on the route reflection clients. From a client's perspective, a route reflector is simply a normal IBGP peer. Any BGP version 4 speaker can be a reflector client.
- It is necessary to export routes from the local AS into the local AS when acting as a route reflector.

  To accomplish this, routers GSR10 and GSR11 have the following line in their configuration files:

```
ip-router policy redistribute from-proto bgp source-as 64901 to-
     proto bgp target-as 64901
```

- If the cluster ID is changed, all BGP sessions with reflector clients will be dropped and restarted.

Chapter 11

Routing Policy Configuration Guide

Route Import and Export Policy Overview

The GSR family of routers supports extremely flexible routing policies. The GSR 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 GSR 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 GSR 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 GSR 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.

Table 5. Default Preference Values

Preference	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 GSR 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 the 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 the 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 the 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. To make key management easier, the concept of a *key-chain* was introduced. Each key-chain has an identifier and can contain up to two keys. One key 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 Configure mode, instead of specifying the key for each interface (which can be up to 16 characters long), you can specify a key-chain identifier.

The GSR supports MD5 specification of OSPF RFC 2178 which uses the MD5 algorithm and an authentication key of up to 16 characters. Thus there are now three authentication schemes available per interface: none, simple and RFC 2178 OSPF MD5 authentication. It is possible to configure different authentication schemes on different interfaces.

RFC 2178 allows multiple MD5 keys per interface. Each key has two times associated with the key:

- A time period that the key will be generated
- A time period that the key will be accepted

The GSR only allows one MD5 key per interface. Also, there are no options provided to specify the time period during which the key would be generated and accepted; the specified MD5 key is always generated and accepted. Both these limitations would be removed in a future release.

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 may fulfill 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 can be **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 can be **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 the 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 GSR 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 17 on page 160.

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 18 on page 164.

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 & 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 & Static Routes to OSPF

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

In the configuration shown in Figure 18 on page 164, suppose 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 GSR 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 **ip-router 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 identifier 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 **ip-router 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 identifier 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 **ip-router 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 identifier 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

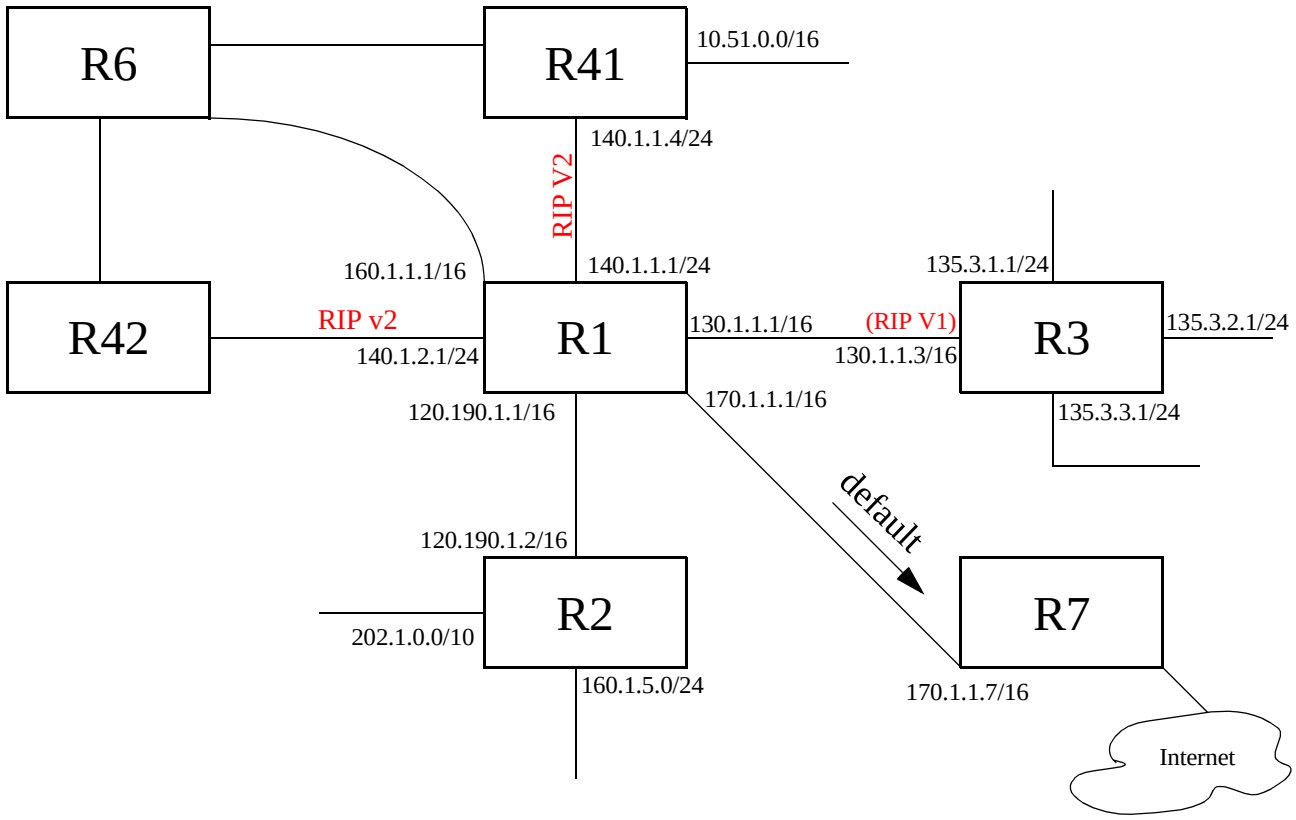
Example 1: Importing from RIP

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 17 on page 160.

Figure 17. 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 RIP Trusted Gateway

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

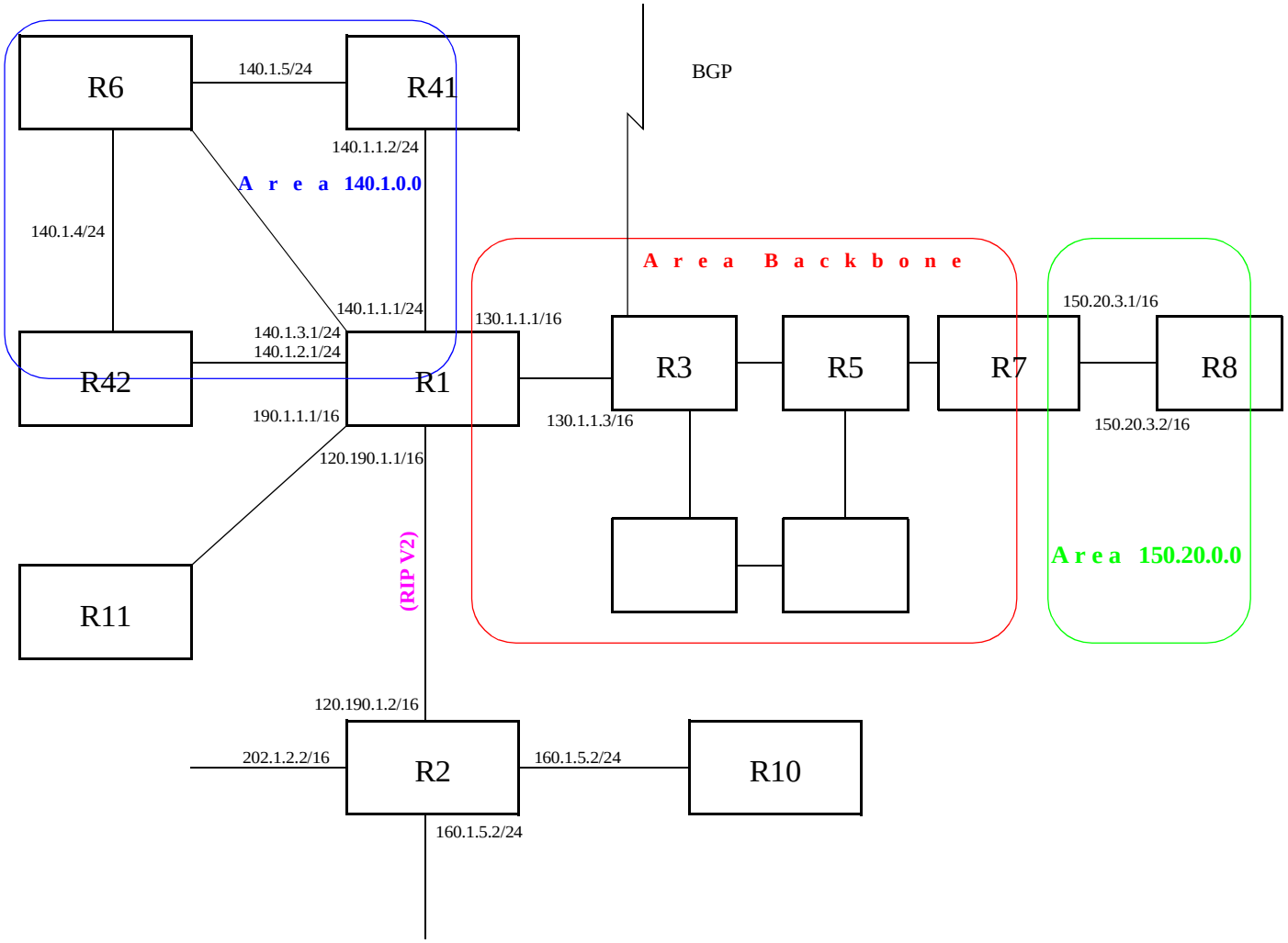
Due to the nature of OSPF, only the importation of ASE routes may be controlled. OSPF intra-and inter-area routes are always imported into the GSR routing table with a preference of 10. If a tag is specified, the import clause will only apply to routes with the specified tag.

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 18 on page 164.

Figure 18. 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 17 on page 160.

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 17 on page 160, 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 GSR routing table into OSPF. It is only possible to export from the GSR 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 18 on page 164.

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 & 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 & Static Routes to OSPF

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

In the configuration shown in Figure 18 on page 164, suppose 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 12

Multicast Routing Configuration Guide

IP Multicast Overview

Multicast routing on the GSR 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 GSRs.

This chapter:

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

IGMP Overview

The GSR 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 GSR, 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.

The GSR 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 GSR will wait for host responses to IGMP queries. The GSR can be configured to deny or accept group membership filters.

DVMRP Overview

DVMRP is an IP multicast routing protocol. On the GSR, 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 GSR's implementation of DVMRP supports the following:

- The mtrace utility, which racks 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. This is known as an (S,G) pair. RPM permits the GSR to maintain multiple (S,G) pairs.

On the GSR, 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 multicast routing protocols. IGMP starts and stops automatically with DVMRP. The GSR supports up to 64 multicast interfaces.

To support backward compatibility on DVMRP interfaces, you can configure the router expire time and prune time on each GSR 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 GSR can send DVMRP data. Scopes can include only addresses of devices on a company's internal network and cannot include addresses that require the GSR to send DVMRP data on the Internet. The GSR also allows control of routing information exchange with peers through route filter rules.

You can also configure tunnels on GSR 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. The GSR supports a maximum of eight tunnels.

Note: Tunnel traffic is not optimized on a per-port basis, and it goes to all ports on an interface, even though IGMP keeps per-port membership information. This is done to minimize CPU overload for tunneled traffic.

Configuring IGMP

You configure IGMP on the GSR 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
- Start the multicast routing protocol (i.e., DVMRP)

Configuring IGMP on an IP Interface

By default IGMP is disabled on the GSR.

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 GSR with a different IGMP Host Membership Query time interval. The interval you set applies to all ports on the GSR. 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 GSR 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 GSR. 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 GSR 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 GSR 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 GSR.

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*> | <*interface-name*>

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 (2 hours).

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 & 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. The following are conventional guidelines for assigning TTL values to a multicast application and their corresponding GSR 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 GSR 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 GSR 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 GSR then sends and receives multicast packets over the tunnel. Tunnels are CPU-intensive; they are not switched directly through the GSR's multitasking ASICs.

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. The GSR supports a maximum of eight tunnels.

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 & DVMRP

You can monitor IGMP and DVMRP information on the GSR.

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 GSR 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 13

IP Policy-Based Forwarding Configuration Guide

Overview

You can configure the GSR to route IP packets according to policies that you define. IP-policy-based routing allows network managers to engineer traffic to make the most efficient use of their network resources.

IP policies forward packets based on layer-3 or layer-4 IP header information. You can define IP policies to route packets to a set of next-hop IP addresses based on any combination the following IP header fields:

- IP protocol
- Source IP address
- Destination IP address
- Source Socket
- Destination Socket
- Type of service

For example, you can set up an IP policy to send packets originating from a certain network through a firewall, while letting other packets bypass the firewall. Using IP policies, sites that have multiple Internet service providers can cause user groups to use different ISPs. You can also create IP policies to select service providers based on various traffic types.

Other uses for IP policy routing include transparent web caching, where all HTTP requests are directed to a local cache server, saving WAN access bandwidth and costs. An ISP can use policy-based routing on an access router to supply high-priority customers with premium levels of service.

Configuring IP Policies

To implement an IP policy, you first create a profile for the packets to be forwarded using an IP policy. For example, you can create a profile defined as "all telnet packets going from network 9.1.0.0/16 to network 15.1.0.0/16". You then associate the profile with an IP policy. The IP policy specifies what to do with the packets that match the profile. For example, you can create an IP policy that sends packets matching a given profile to next-hop gateway 100.1.1.1.

Configuring an IP policy consists of the following tasks:

- Defining a profile
- Associating the profile with a policy
- Applying the IP policy to an interface

Defining an ACL Profile

An ACL profile specifies the criteria packets must meet to be eligible for IP policy routing. You define profiles with the **acl** command. For IP policy routing, the GSR uses the packet-related information from the **acl** command and ignores the other fields.

For example, the following **acl** command creates a profile called "prof1" for telnet packets going from network 9.1.1.5 to network 15.1.1.2:

```
gs/r(config)# acl prof1 permit ip 9.1.0.0/16 15.1.0.0/16 any any telnet
0
```

See the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual* for complete syntax information for the **acl** command.

Note: ACLs for non-IP protocols cannot be used for IP policy routing.

Associating the Profile with an IP Policy

Once you have defined a profile with the **acl** command, you associate the profile with an IP policy by entering one or more **ip-policy** statements. An **ip-policy** statement specifies the next-hop gateway (or gateways) where packets matching a profile are forwarded. To cause packets matching a defined profile to be forwarded to a next-hop gateway, enter the following command in Configure mode:

Forward packets matching a profile to a next-hop gateway.	**ip-policy** *<name>* **permit acl** *<profile>* **next-hop-list** *<ip-addr-list>*

For example, the following command creates an IP policy called "p1" and specifies that packets matching profile "prof1" are forwarded to next-hop gateway 10.10.10.10:

```
gs/r(config)# ip-policy p1 permit acl prof1 next-hop-list 10.10.10.10
```

You can also set up a policy to prevent packets from being forwarded by an IP policy. To prevent packets matching a defined profile from being forwarded by an IP policy to a next-hop gateway, enter the following command in Configure mode:

Prevent packets matching a profile from being forwarded by an IP policy.	**ip-policy** *<name>* **deny acl** *<profile>*

Packets matching the specified profile are forwarded using dynamic routes instead.

For example, the following command creates an IP policy called "p2" that prevents packets matching prof1 from being forwarded using an IP policy:

```
gs/r(config)# ip-policy p2 deny acl prof1
```

Creating Multi-statement IP Policies

An IP policy can contain more than one **ip-policy** statement. For example, an IP policy can contain one statement that sends all packets matching a profile to one next-hop gateway, and another statement that sends packets matching a different profile to a different next-hop gateway. If an IP policy has multiple **ip-policy** statements, you can assign each statement a sequence number that controls the order in which they are evaluated. Statements are evaluated from lowest sequence number to highest.

To specify the order in which IP policy statements are evaluated by an IP policy, enter the following command in Configure mode:

Specify a sequence number for IP policy statements	**ip-policy** *<name>* **permit|deny acl** *<profile>* **sequence** *<num>*

For example, the following commands create an IP policy called "p3", which consists of two IP policy statements. The **ip policy permit** statement has a sequence number of 1, which means it is evaluated before the **ip policy deny** statement, which has a sequence number of 900.

```
gs/r(config)# ip-policy p3 permit acl prof1 next-hop-list 10.10.10.10
    sequence 1
gs/r(config)# ip-policy p3 deny acl prof2 sequence 900
```

Setting Load Distribution for Next-hop Gateways

You can specify up to four next-hop gateways in an **ip-policy** statement. If you specify more than one next-hop gateway, you can control how the load is distributed among them. You can cause each new flow to use the first available next-hop gateway in the **ip-policy permit** statement, or you can cause flows to use all the next-hop gateways in the **ip-policy permit** statement sequentially.

To set the load distribution for next-hop gateways, enter one of the following commands in Configure mode:

Use the first available next-hop gateway in the **ip-policy permit** statement for all flows. This is the default.	**ip-policy** *<name>* **set load-policy first-available**
Sequentially pick the next gateway in the list for each new flow.	**ip-policy** *<name>* **set load-policy round-robin**
Determine the next hop gateway.	**ip-policy** *<name>* **set load-policy ip-hash sip|dip|both**

Setting the IP Policy Action

You can specify when to apply the IP policy route with respect to dynamic or statically configured routes. The GSR can cause packets to use the IP policy route first, then the dynamic route if the next-hop gateway specified in the IP policy is unavailable; use the dynamic route first, then the IP policy route; or drop the packets if the next-hop gateway specified in the IP policy is unavailable.

To set the IP policy action with respect to dynamic or statically configured routes, enter one of the following commands in Configure mode:

Cause packets matching the profile to use the IP policy route first. If the next-hop gateway is not reachable, use the dynamic route instead.	**ip-policy** <*name*> **permit acl** <*profile*> **action policy-first**
Route packets matching the profile using dynamic routes first. If a dynamic route is not available, then route packets matching the profile using the IP policy gateway.	**ip-policy** <*name*> **permit acl** <*profile*> **action policy-last**
Cause packets matching the profile to use the IP policy route. If the next-hop gateway is not reachable, then drop the packets.	**ip-policy** <*name*> **permit acl** <*profile*> **action policy-only**
Drop packets matching the profile.	**ip-policy** <*name*> **permit acl** <*profile*> **next-hop-list null**
Drop packets that do not match any profile.	**ip-policy** <*name*> **permit everything-else next-hop-list null**

Checking the Availability of Next-hop Gateways

The GSR can check the availability of next-hop gateways by querying them with ICMP_ECHO_REQUESTS. Only gateways that respond to these requests are used for forwarding packets. To configure the GSR to do this, enter the following command in Configure mode:

Periodically check the availability of next-hop gateways.	**ip-policy** <*name*> **set pinger on**

Note: Some hosts may have disabled responding to ICMP_ECHO packets. Make sure each next-hop gateway can respond to ICMP_ECHO packets before using this option.

Applying an IP Policy to an Interface

After you define the IP policy, it must be applied to an inbound IP interface. Once the IP policy is applied to the interface, packets start being forwarded according to the IP policy. To apply an IP policy to an interface, enter one of the following commands in Configure mode:

Apply a defined IP policy to an IP interface.	**ip-policy** <*name*> **apply interface** <*InterfaceName*>
Apply a defined IP policy to all IP interfaces on the GSR.	**ip-policy** <*name*> **apply interface all**

Applying an IP Policy to Locally Generated Packets

You can apply an IP policy to locally generated packets (that is, packets generated by the GSR). To do this, enter the following command in Configure mode:

Cause packets generated by the GSR to be forwarded according to an IP policy.	**ip-policy** <*name*> **apply local**

IP Policy Configuration Examples

This section presents some examples of IP policy configurations. The following uses of IP policies are demonstrated:

- Routing traffic to different ISPs
- Prioritizing service to customers
- Authenticating users through a firewall
- Firewall load balancing

Routing Traffic to Different ISPs

Sites that have multiple Internet service providers can create IP policies that cause different user groups to use different ISPs. You can also create IP policies to select service providers based on various traffic types.

In the sample configuration in Figure 19, the policy router is configured to divide traffic originating within the corporate network between different ISPs (100.1.1.1 and 200.1.1.1).

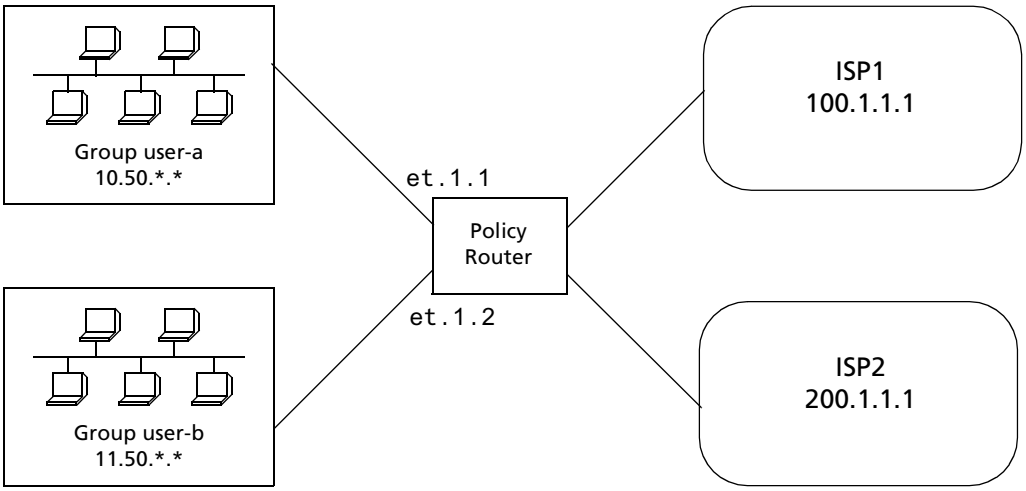


Figure 19. Using an IP policy to route traffic to two different ISPs

HTTP traffic originating from network 10.50.0.0 for destination 207.31.0.0/16 is forwarded to 100.1.1.1. Non-HTTP traffic originating from network 10.50.0.0 for destination 207.31.0.0/16 is forwarded to 200.1.1.1. All other traffic is forwarded to 100.1.1.1.

The following is the IP policy configuration for the Policy Router in Figure 19:

```
interface create ip user-a address-netmask 10.50.1.1/16 port et.1.1
interface create ip user-b address-netmask 11.50.1.1/16 port et.1.2

acl user-a-http  permit ip 10.50.0.0/16 207.31.0.0/16 any http 0
acl user-a  permit ip 10.50.0.0/16 207.31.0.0/16 any any 0
acl user-b  permit ip 11.50.0.0/16 any any any 0

ip-policy net-a permit acl user-a-http next-hop-list 100.1.1.1 action
policy-first sequence 20

ip-policy net-a permit acl user-a next-hop-list 200.1.1.1 action policy-
only sequence 25

ip-policy net-a apply interface user-a

ip-policy net-b permit acl user-b next-hop-list 200.1.1.1 action policy-
first

ip-policy net-b apply interface user-b
```

Prioritizing Service to Customers

An ISP can use policy-based routing on an access router to supply different customers with different levels of service. The sample configuration in Figure 20 shows a GSR using an IP policy to classify customers and route traffic to different networks based on customer type.

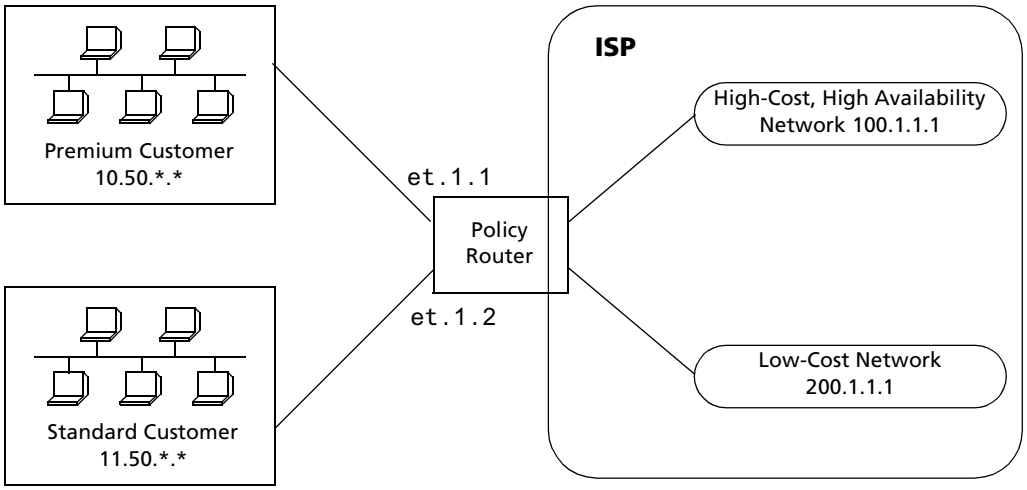


Figure 20. Using an IP policy to prioritize service to customers

Traffic from the premium customer is load balanced across two next-hop gateways in the high-cost, high-availability network. If neither of these gateways is available, then packets are forwarded based on dynamic routes learned via routing protocols.

Traffic from the standard customer always uses one gateway (200.1.1.1). If for some reason that gateway is not available, packets from the standard customer are dropped.

The following is the IP policy configuration for the Policy Router in Figure 20:

```
interface create ip premium-customer address-netmask 10.50.1.1/16 port
et.1.1

interface create ip standard-customer address-netmask 11.50.1.1/16 port
et.1.2

acl premium-customer permit ip 10.50.0.0/16 any any any 0
acl standard-customer   permit ip 11.50.0.0/16 any any any 0

ip-policy p1 permit acl premium-customer next-hop-list "100.1.1.1
100.1.1.2" action policy-first sequence 20

ip-policy apply interface premium-customer

ip-policy p2 permit acl standard-customer next-hop-list 200.1.1.1
action policy-only sequence 30

ip-policy apply interface standard-customer
```

Authenticating Users Through a Firewall

You can define an IP policy that authenticates packets from certain users via a firewall before accessing the network. If for some reason the firewall is not responding, the packets to be authenticated are dropped. Figure 21 illustrates this kind of configuration.

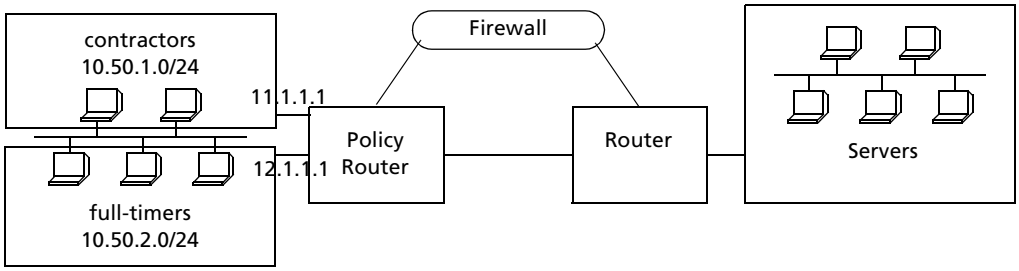


Figure 21. Using an IP policy to authenticate users through a firewall

Packets from users defined in the “contractors” group are sent through a firewall. If the firewall cannot be reached packets from the contractors group are dropped. Packets from users defined in the “full-timers” group do not have to go through the firewall.

The following is the IP policy configuration for the Policy Router in Figure 21:

```
interface create ip mls0 address-netmask 10.50.1.1/16 port et.1.1

acl contractors permit ip 10.50.1.0/24 any any any 0
acl full-timers permit ip 10.50.2.0/24 any any any 0

ip-policy access permit acl contractors next-hop-list 11.1.1.1 action
policy-only
ip-policy access permit acl full-timers next-hop-list 12.1.1.1 action
policy-first
ip-policy access apply interface mls0
```

Firewall Load Balancing

The next hop gateway can be selected by the following information in the IP packet: source IP, destination IP, or both the source and destination IP. Figure 22 illustrates this configuration.

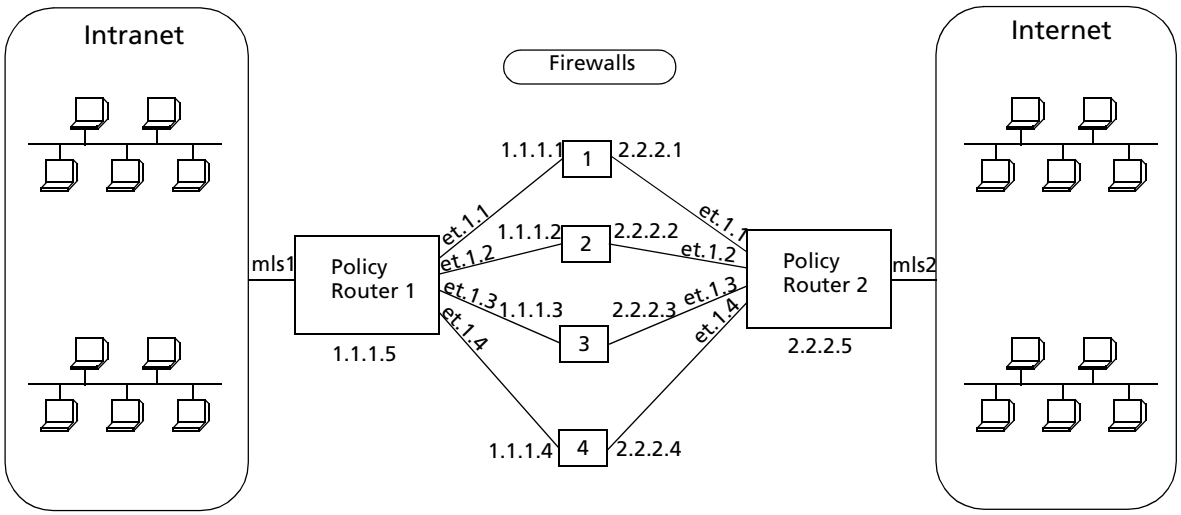


Figure 22. Selecting Next Hop Gateway from IP Packet Information

One session should always go to a particular firewall for persistence.

The following is the configuration for Policy Router 1 in Figure 22.

```
vlan create firewall
vlan add ports et.1.(1-5) to firewall

interface create ip firewall address-netmask 1.1.1.5/16 vlan firewall

acl firewall permit ip any any any 0

ip-policy p1 permit acl firewall next-hop-list “1.1.1.1 1.1.1.2 1.1.1.3
1.1.1.4” action policy-only

ip-policy p1 set load-policy ip-hash both

ip-policy p1 apply interface mls1
```

The following is the configuration for Policy Router 2 in Figure 22.

```
vlan create firewall
vlan add ports et.1.(1-5) to firewall

interface create ip firewall address-netmask 2.2.2.5/16 vlan firewall

acl firewall permit ip any any any 0

ip-policy p2 permit acl firewall next-hop-list “2.2.2.1 2.2.2.2 2.2.2.3
2.2.2.4” action policy-only

ip-policy p2 set load-policy ip-hash both

ip-policy p2 apply interface mls2
```

Monitoring IP Policies

The **ip-policy show** command reports information about active IP policies, including profile definitions, policy configuration settings, and next-hop gateways. The command also displays statistics about packets that have matched an IP policy statement as well as the number of packets that have been forwarded to each next-hop gateway.

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

Display information about all IP policies.	**ip-policy show all** **ip-policy show policy-name all**
Display statistics about a specific IP policy.	**ip-policy show policy-name** <*name*>
Display information about all IP policies on a specified interface.	**ip-policy show interface** <*interface*>
Display information about IP policies that have been applied to all interfaces	**ip-policy show interface all**
Clear statistics gathered for IP policies.	**ip-policy clear all|policy-name** <*name*>**|all**

For example, to display information about an active IP policy called "p1", enter the following command in Enable mode:

```
gs/r# ip-policy show policy-name p1

------------------------------------------------------------------------------
IP Policy name     : p1 ❶
Applied Interfaces : int1  ❷
Load Policy        : first available  ❸

❹               ❺                ❻               ❼         ❽         ❾  ❿
ACL          Source IP/Mask     Dest. IP/Mask      SrcPort   DstPort   TOS Prot
---          --------------     -------------      --------- --------- --- ----
prof1        9.1.1.5/32         15.1.1.2           any       any         0 IP
prof2        2.2.2.2/32         anywhere           any       any         0 IP
everything   anywhere           anywhere           any       any         0 IP

                                                      Next Hop Information
                                                      --------------------
 ⓫    ⓬      ⓭          ⓮    ⓯                          ⓰             ⓱   ⓲
Seq   Rule   ACL        Cnt Action                     Next Hop        Cnt Last
---   ----   --------   --- -----------                --------        --- ----
10    permit prof1      0   Policy Only                11.1.1.2        0   Dwn
20    permit prof2      0   Policy Last                1.1.1.1         0   Dwn
                                                      2.2.2.2         0   Dwn
                                                      3.3.3.3         0   Dwn
999   permit everything 0   Policy Only                drop            N/A N/A
65536 deny   deny       0   N/A                        normal fwd      N/A N/A
㉑
```

Legend:

1. The name of the IP policy.
2. The interface where the IP policy was applied.
3. The load distribution setting for IP-policy statements that have more than one next-hop gateway; either first available (the default) or round-robin.
4. The names of the profiles (created with an **acl** statement) associated with this IP policy.
5. The source address and filtering mask of this flow.
6. The destination address and filtering mask of this flow.
7. For TCP or UDP, the number of the source TCP or UDP port.
8. For TCP or UDP, the number of the destination TCP or UDP port.
9. The TOS value in the packet.
10. IP protocol (ICMP, TCP UDP).
11. The sequence in which the statement is evaluated. IP policy statements are listed in the order they are evaluated (lowest sequence number to highest).
12. The rule to apply to the packets matching the profile: either permit or deny
13. The name of the profile (ACL) of the packets to be forwarded using an IP policy.
14. The number of packets that have matched the profile since the IP policy was applied (or since the **ip-policy clear** command was last used)
15. The method by which IP policies are applied with respect to dynamic or statically configured routes; possible values are Policy First, Policy Only, or Policy Last.
16. The list of next-hop gateways in effect for the policy statement.
17. The number of packets that have been forwarded to this next-hop gateway.
18. The state of the link the last time an attempt was made to forward a packet; possible values are up, dwn, or N/A.
19. Implicit deny rule that is always evaluated last, causing all packets that do not match one of the profiles to be forwarded normally (with dynamic routes).

Chapter 14

Network Address Translation Configuration Guide

Overview

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

Network Address Translation (NAT) allows an IP address used within one network to be translated into a different IP address used within another network. NAT is often used to map addresses used in a private, local intranet to one or more addresses used in the public, global Internet. NAT provides the following benefits:

- Limits the number of IP addresses used for private intranets that are required to be registered with the Internet Assigned Numbers Authority (IANA).
- Conserves the number of global IP addresses needed by a private intranet (for example, an entity can use a single IP address to communicate on the Internet).
- Maintains privacy of local networks, as internal IP addresses are hidden from public view.

With NAT, the local network is designated the *inside* network and the global Internet is designated the *outside* network. In addition, the GSR supports Port Address Translation (PAT) for either static or dynamic address bindings.

The GSR allows you to create the following NAT address bindings:

- Static, one-to-one binding of inside, local address or address pool to outside, global address or address pool. A static address binding does not expire until the command that defines the binding is negated. IP addresses defined for static bindings cannot be reassigned. For static address bindings, PAT allows TCP or UDP port numbers to be translated along with the IP addresses.
- Dynamic binding between an address from a pool of local addresses to an address from a pool of outside addresses. With dynamic address binding, you define local and global address pools from which the addresses bindings can be made. IP addresses defined for dynamic binding are reassigned whenever they become free. For dynamic address bindings, PAT allows port address translation if no addresses are available from the global address pool. PAT allows port address translation for each address in the global pool. The ports are dynamically assigned between the range of 1024 to 4999. Hence, you have about 4,000 ports per global IP address.

  Dynamic bindings are removed automatically when the flow count goes to zero. At this point, the corresponding port (if PAT enabled) or the global IP address is freed and can be reused the next time. Although there are special cases like FTP where the flows are not installed for the control path, the binding will be removed only by the dynamic binding timeout interval.

Configuring NAT

The following are the steps in configuring NAT on the GSR:

1. Setting the NAT interfaces to be “inside” or “outside.”
2. Setting the NAT rules (static or dynamic).

Setting Inside and Outside Interfaces

When NAT is enabled, address translation is only applied to those interfaces which are defined to NAT as “inside” or “outside” interfaces. NAT only translates packets that arrive on a defined inside or outside interface.

To specify an interface as inside (local) or outside (global), enter the following command in Configure mode.

Define an interface as inside or outside for NAT.	**nat set interface** *<InterfaceName>* **inside|outside**

Setting NAT Rules

Static

You create NAT static bindings by entering the following command in Configure mode.

Enable NAT with static address binding.	**nat create static protocol ip|tcp|udp local-ip** *<local-ip-add/address range>* **global-ip** *<global-ip-add/address range>* **[local-port** *<tcp/udp local-port>***|any] [global-port** *<tcp/udp global-port>***|any]**

Dynamic

You create NAT dynamic bindings by entering the following command in Configure mode:.

Enable NAT with dynamic address binding.	**nat create dynamic local-acl-pool** *<local-acl>* **global-pool** *<ip-addr/ip-addr-range/ip-addr-list>* **[matches-interface** *<interface>***] [enable-ip-overload]**

For dynamic address bindings, you define the address pools with previously-created ACLs. You can also specify the **enable-port-overload** parameter to allow PAT.

Managing Dynamic Bindings

As mentioned previously, dynamic address bindings expire only after a period of non-use or when they are manually deleted. The default timeout for dynamic address bindings is 1440 minutes (24 hours). You can manually delete dynamic address bindings for a specific address pool or delete all dynamic address bindings.

To set the timeout for dynamic address bindings, enter the following command in Configure mode.

Set timeout for dynamic address bindings.	**nat set dynamic-binding-timeout** *<minutes>* **|disable**

To flush dynamic address bindings, enter the following command in Enable mode.

Flush dynamic address bindings.	**nat flush-dynamic-binding all|pool-specified [local-acl-pool** *<local-acl>***] [global-pool** *<ip-addr/address range>***]**

NAT and FTP

File Transfer Protocol (FTP) packets require special handling with NAT, because the FTP PORT command packets contain IP address information within the data portion of the packet. It is therefore important for NAT to know which control port is used for FTP (the default is port 21) and the timeout for the FTP session (the default is 30 minutes). If FTP packets will arrive on a different port number, you need to specify that port to NAT.

To define FTP parameters to NAT, enter the following commands in Configure mode.

Specify the FTP control port.	**nat set ftp-control-port** <*port number*>
Specify the FTP session timeout.	**nat set ftp-session-timeout** <*minutes*>

Monitoring NAT

To display NAT information, enter the following command in Enable mode.

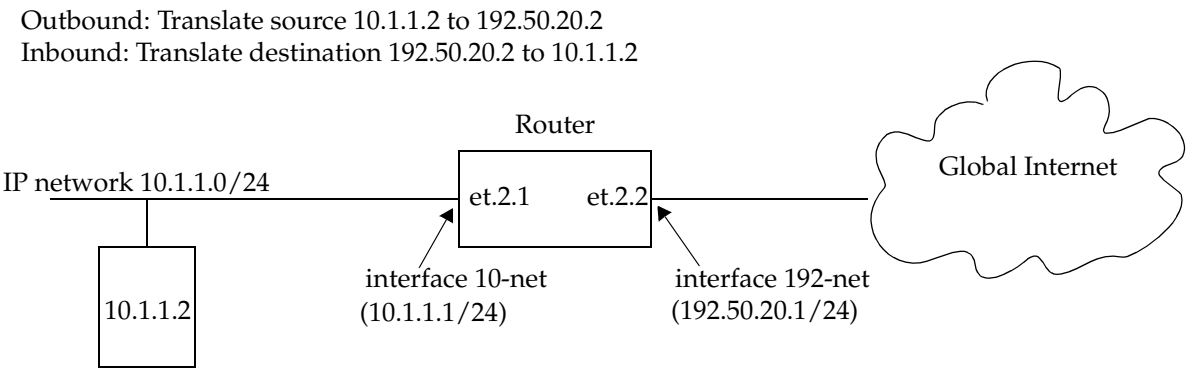
Display NAT information.	**nat show [translations all|**<*type*>**]** **[timeouts] [statistics]**

Configuration Examples

This section shows examples of NAT configurations.

Static Configuration

The following example configures a static address binding for inside address 10.1.1.2 to outside address 192.50.20.2:



The first step is to create the interfaces:

```
interface create ip 10-net address-netmask 10.1.1.1/24 port et.2.1
interface create ip 192-net address-netmask 192.50.20.1/24 port et.2.2
```

Next, define the interfaces to be NAT “inside” or “outside”:

```
nat set interface 10-net inside
nat set interface 192-net outside
```

Then, define the NAT static rules:

```
nat create static protocol ip local-ip 10.1.1.2 global-ip 192.50.20.2
```

Using Static NAT

Static NAT can be used when the local and global IP addresses are to be bound in a fixed manner. These bindings never get removed nor time out until the static NAT command itself is negated. Static binding is recommended when you have a need for a permanent type of binding.

The other use of static NAT is when the out to in traffic is the first to initialize a connection, i.e., the first packet is coming from outside to inside. This could be the case when you have a server in the local network and clients located remotely. Dynamic NAT would not work for this case as bindings are always created when an in to out Internet connection occurs. A typical example is a web server inside the local network, which could be configured as follows:

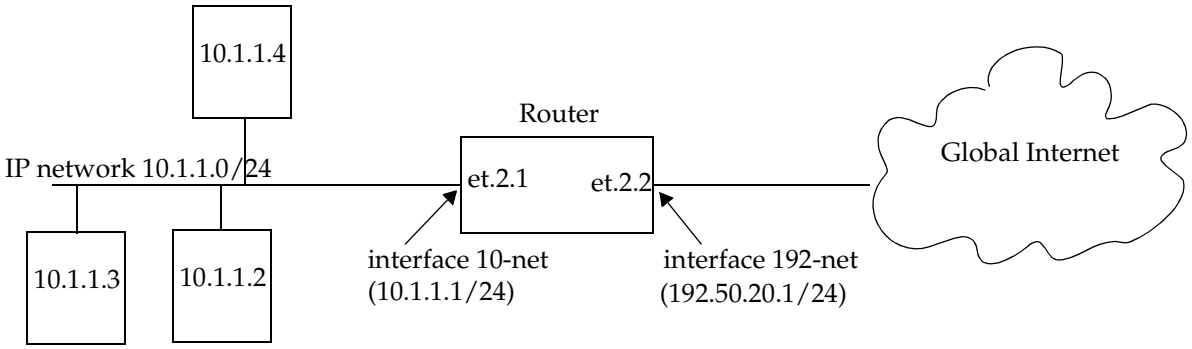
```
nat create static protocol tcp local-ip 10.1.1.2 global-ip 192.50.20.2
local-port 80 global-port 80
```

This server, 10.1.1.2, is advertised as 192.50.20.2 to the external network.

Dynamic Configuration

The following example configures a dynamic address binding for inside addresses 10.1.1.0/24 to outside address 192.50.20.0/24:

Outbound: Translate source pool 10.1.1.0/24 to global pool 192.50.20.0/24



The first step is to create the interfaces:

```
interface create ip 10-net address-netmask 10.1.1.1/24 port et.2.1
interface create ip 192-net address-netmask 192.50.20.1/24 port et.2.2
```

Next, define the interfaces to be NAT "inside" or "outside":

```
nat set interface 10-net inside
nat set interface 192-net outside
```

Then, define the NAT dynamic rules by first creating the source ACL pool and then configuring the dynamic bindings:

```
acl lcl permit ip 10.1.1.0/24
nat create dynamic local-acl-pool lcl global-pool 192.50.20.0/24
```

Using Dynamic NAT

Dynamic NAT can be used when the local network (inside network) is going to initialize the connections. It creates a binding at run time when a packet is sent from a local network, as defined by the NAT dynamic local ACl pool. The network administrator does not have to worry about the way in which the bindings are created; the network administrator just sets the pools and the GSR automatically chooses a free global IP from the global pool for the local IP.

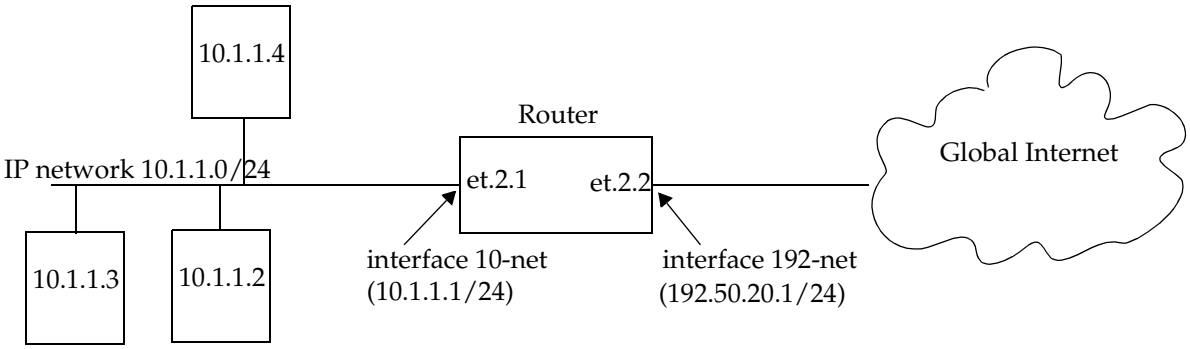
Dynamic bindings are removed when the flow count for that binding goes to zero or the timeout has been reached. The free globals are used again for the next packet.

A typical problem is that if there are more local IP addresses as compared to global IP addresses in the pools, then packets will be dropped if all the globals are used. A solution to this problem is to use PAT with NAT dynamic. This is only possible with TCP or UDP protocols.

Dynamic NAT with IP Overload (PAT) Configuration

The following example configures a dynamic address binding for inside addresses 10.1.1.0/24 to outside address 192.50.20.0/24:

Outbound: Translate source pool 10.1.1.0/24 to global pool 192.50.20.1-192.50.20.3



The first step is to create the interfaces:

```
interface create ip 10-net address-netmask 10.1.1.1/24 port et.2.1
interface create ip 192-net address-netmask 192.50.20.1/24 port et.2.2
```

Next, define the interfaces to be NAT “inside” or “outside”:

```
nat set interface 10-net inside
nat set interface 192-net outside
```

Then, define the NAT dynamic rules by first creating the source ACL pool and then configuring the dynamic bindings:

```
acl lcl permit ip 10.1.1.0/24
nat create dynamic local-acl-pool lcl global-pool 192.50.20.1-192.50.20.3
```

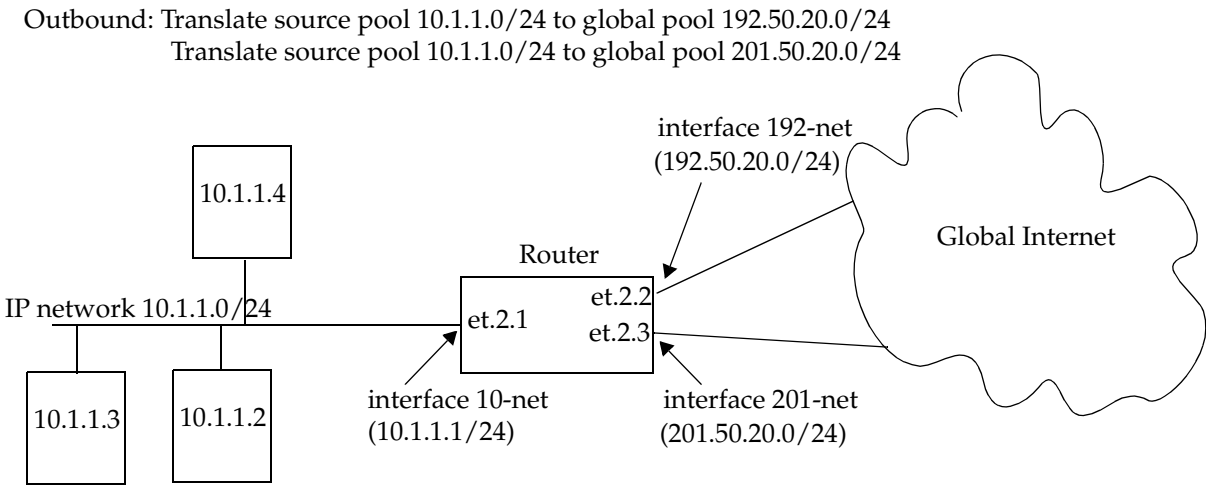
Using Dynamic NAT with IP Overload

Dynamic NAT with IP overload can be used when the local network (inside network) will be initializing the connections using TCP or UDP protocols. It creates a binding at run time when the packet comes from a local network defined in the NAT dynamic local ACL pool. The difference between the dynamic NAT and dynamic NAT with PAT is that PAT uses port (layer 4) information to do the translation. Hence, each global IP has about 4000 ports that can be translated. NAT on the GSR uses the standard BSD range of ports from 1024-4999 which is fixed and cannot be configured by the user. The network administrator does not have to worry about the way in which the bindings are created; he/she just sets the pools and the GSR automatically chooses a free global IP from the global pool for the local IP.

Dynamic bindings are removed when the flow count goes to zero or the timeout has been reached. The removal of bindings frees the port for that global and the port is available for reuse. When all the ports for that global are used, then ports are assigned from the next free global. If no more ports and globals are available, the packets will be dropped.

Dynamic NAT with Outside Interface Redundancy

The following example configures a dynamic address binding for inside addresses 10.1.1.0/24 to outside addresses 192.50.20.0/24 on interface 192-net and to outside addresses 201.50.20.0/24 on interface 201-net:



The first step is to create the interfaces:

```
interface create ip 10-net address-netmask 10.1.1.1/24 port et.2.1
interface create ip 192-net address-netmask 192.50.20.0/24 port et.2.2
interface create ip 201-net address-netmask 201.50.20.0/24 port et.2.3
```

Next, define the interfaces to be NAT “inside” or “outside”:

```
nat set interface 10-net inside
nat set interface 192-net outside
nat set interface 201-net outside
```

Then, define the NAT dynamic rules by first creating the source ACL pool and then configuring the dynamic bindings:

```
acl lcl permit ip 10.1.1.0/24
nat create dynamic local-acl-pool lcl global-pool 192.50.20.0/24 matching-
if 192-net
nat create dynamic local-acl-pool lcl global-pool 210.50.20.0/24 matching-
if 201-net
```

Using Dynamic NAT with Matching Interface Redundancy

If you have redundant connections to the remote network via two different interfaces, you can use NAT for translating the local address to the different global pool specified for the two connections. This case is possible when you have two ISPs connected on two different interfaces to the Internet. Through a routing protocol, some routes will result in traffic going out of one interface and for others going out on the other interface. NAT will check which interface the packet is going out from before selecting a global pool. Hence, you can specify two different global pools with the same local ACL pool on two different interfaces.

Chapter 15

Web Hosting Configuration Guide

Overview

Accessing information on Web sites for both work or personal purposes is becoming a normal practice for an increasing number of people. For many companies, fast and efficient Web access is important for both external customers who need to access the company Web sites, as well as for users on the corporate intranet who need to access Internet Web sites.

The following features on the GSR provide ways to improve Web access for external and internal users:

- Load balancing allows incoming HTTP requests to a company's Web site to be distributed across several physical servers. If one server should fail, other servers can pick up the workload.
- Web caching allows HTTP requests from internal users to Internet sites to be redirected to cached Web objects on local servers. Not only is response time faster since requests can be handled locally, but overall WAN bandwidth usage is reduced.

Note: Load balancing and web caching can be performed using application software, however, the GSR can perform these function much faster as the redirection is handled at lower levels.

Load Balancing

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

You can use the load balancing feature on the GSR to distribute session load across a group of servers. If you configure the GSR to provide load balancing, client requests that go through the GSR can be redirected to any one of several predefined hosts. With load balancing, clients access servers through a virtual IP. The GSR transparently redirects the requests with no change required on the clients or servers; all configuration and redirection is done on the GSR.

Configuring Load Balancing

The following are the steps in configuring load balancing on the GSR:

1. Create a logical group of load balancing servers and define a virtual IP for the group.
2. Specify the policy for distributing workload for this group of load balancing servers. This step is optional; by default, the GSR assigns sessions to servers in a round-robin (sequential) manner.
3. Define the servers in the group.

Creating the Server Group

To use load balancing, you create a logical group of load balancing servers and define a virtual IP for the server that the clients will use to access the server pool.

To create the server group and define the virtual IP for the server, enter the following command in Configure mode:.

Create group of load balancing servers.	**load-balance create group-name** *<group name>* **virtual-ip** *<ipaddr>* **virtual-port** *<port number>* **protocol tcp|udp**
Create a range of load balancing server groups.	**load-balance create vip-range-name** *<range name>* **vip-range** *<range>* **virtual-port** *<port number>* **protocol tcp|udp**

Specifying Load Balancing Policy (Optional)

The default policy for distributing workload among the load balancing servers is "round-robin," where the GSR selects the server on a rotating basis without regard to the load on individual servers. Other policies can be chosen for the group, including least loaded, where the server with the fewest number of sessions bound to it is selected to service a new session. The weighted round robin policy is a variation of the round-robin policy, where each server takes on new sessions according to its assigned weight. If you choose the weighted round robin policy, you must assign a weight to each server that you add to the load balancing group.

To specify the load balancing policy, enter the following command in Configure mode:.

Specify load balancing policy.	**load-balance set policy-for-group** *<group name>* **policy** *<policy>*

Adding Servers to the Load Balancing Group

Once a logical server group is created, you specify the servers that can handle client requests. When the GSR receives a client request directed to the virtual server address, it redirects the request to the actual server address and port. Server selection is done according to the specified policy.

To add servers to the server group, enter the following command in Configure mode:.

Add load balancing servers to a specific server group.	**load-balance add host-to-group** *<ipaddr/range>* **group-name** *<group name>* **port** *<port number>* **[weight** *<weight>***]**
Add range of load balancing servers to a range of server groups.	**load-balance add host-to-vip-range** *<range>* **vip-range-name** *<range name>* **port** *<port number>* **[weight** *<weight>***]**

Setting Server Status

It may become necessary at times to prevent new sessions from being directed to one or more load balancing servers. For example, if you need to perform maintenance tasks on a server system, you might want new sessions to temporarily *not* be directed to that server. Setting the status of a server to "down" prevents new sessions from being directed to that server. The "down" status does not affect any current sessions on the server. When the server is again ready to accept new sessions, you can set the server status to "up."

To set the status of a load balancing server, enter the following command in Enable mode.

Set status of load balancing server.	**load-balance set server-status server-ip** *<ipaddr/range>* **server-port** *<port number>* **group-name** *<group name>* **status up|down**

Load Balancing and FTP

File Transfer Protocol (FTP) packets require special handling with load balancing, because the FTP PORT command packets contain IP address information within the data portion of the packet. If the FTP control port used is not port 21, it is important for the GSR to know the port number that is used for FTP.

To define an FTP control port (other than port 21) to the load balancing function, enter the following command in Configure mode.

Specify the FTP control port.	**load-balance set ftp-control-port** *<port number>*

Allowing Access to Load Balancing Servers

Load balancing causes both source and destination addresses to be translated on the GSR. It may be undesirable in some cases for a source address to be translated; for example, when data is to be updated on an individual server. Specified hosts can be allowed to directly access servers in the load balancing group without address translation. Note, however, that such hosts cannot use the virtual IP address and port number to access the load balancing group of servers.

To allow specified hosts to access the load balancing servers without address translation, enter the following command in Configure mode.

Specify the hosts that can access servers without address translation.	**load-balance allow access-to-servers client-ip** *<ipaddr/range>* **group-name** *<group name>*

Setting Timeouts for Load Balancing Mappings

A mapping between a host (source) and a load-balancing server (destination) times out after a certain period. You can specify the timeout for source-destination load balancing mappings.

To specify the timeout for load balancing mappings, enter the following command in Configure mode.

Specify the timeout for source-destination mappings.	**load-balance set mappings-age-timer** *<timer>*

Displaying Load Balancing Information

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

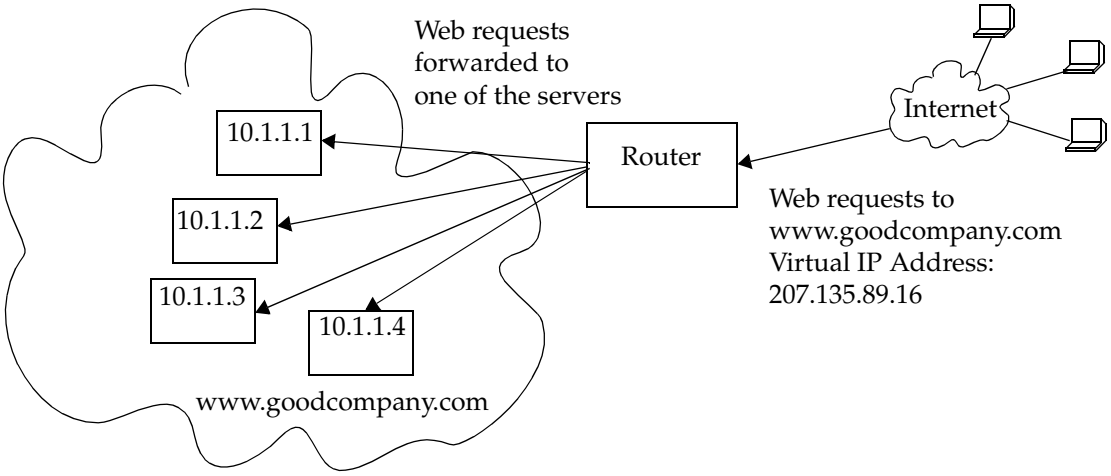
Show the groups of load balancing servers.	**load-balance show virtual-hosts group-name** *<group name>* **virtual-ip** *<ipaddr>* **virtual-port** *<port number>*
Show source-destination bindings.	**load-balance show source-mappings client-ip** *<ipaddr/range>* **virtual-ip** *<ipaddr>* **virtual-port** *<port number>* **destination-host-ip** *<ipaddr>*
Show load balancing statistics.	**load-balance show statistics group-name** *<group name>* **virtual-ip** *<ipaddr>* **virtual-port** *<port number>*
Show load balance hash table statistics.	**load-balance show hash-stats**

Configuration Examples

This section shows examples of load balancing configurations.

Web Hosting with One Virtual Group and Multiple Destination Servers

In the following example, a company web site is established with a URL of www.GoodCompany.com. The system administrator configures the networks so that the GSR forwards web requests among four separate servers, as shown below.



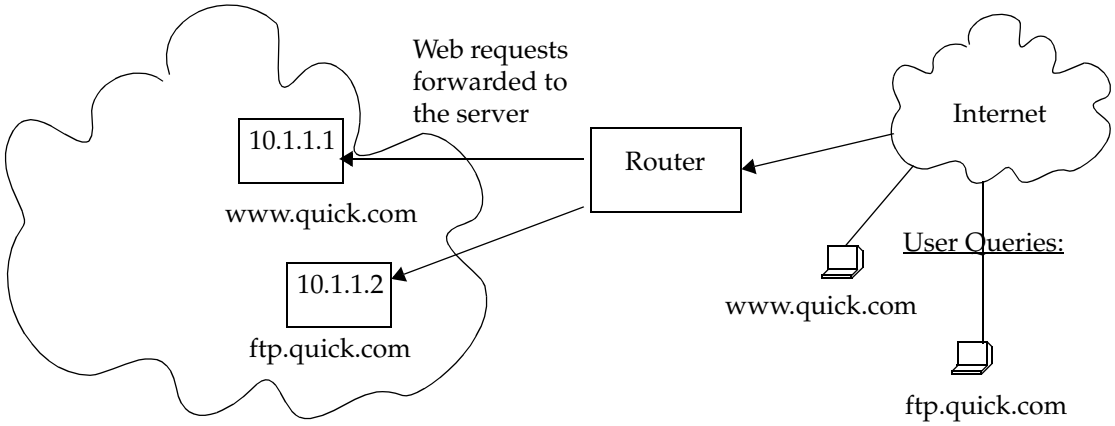
Domain Name	Virtual IP	TCP Port	Real Server IP	TCP Port
www.goodcompany.com	207.135.89.16	80	10.1.1.1	80
			10.1.1.2	80
			10.1.1.3	80
			10.1.1.4	80

The network shown above can be created with the following load-balance commands:

```
load-balance create group-name goodcompany-www virtual-ip 207.135.89.16 virtual
port 80 protocol tcp
load-balance add host-to-group 10.1.1.1-10.1.1.4 group-name goodcompany-www port
80
```

Web Hosting with Multiple Virtual Groups and Multiple Destination Servers

In the following example, two different servers are used to provide different services for a site.



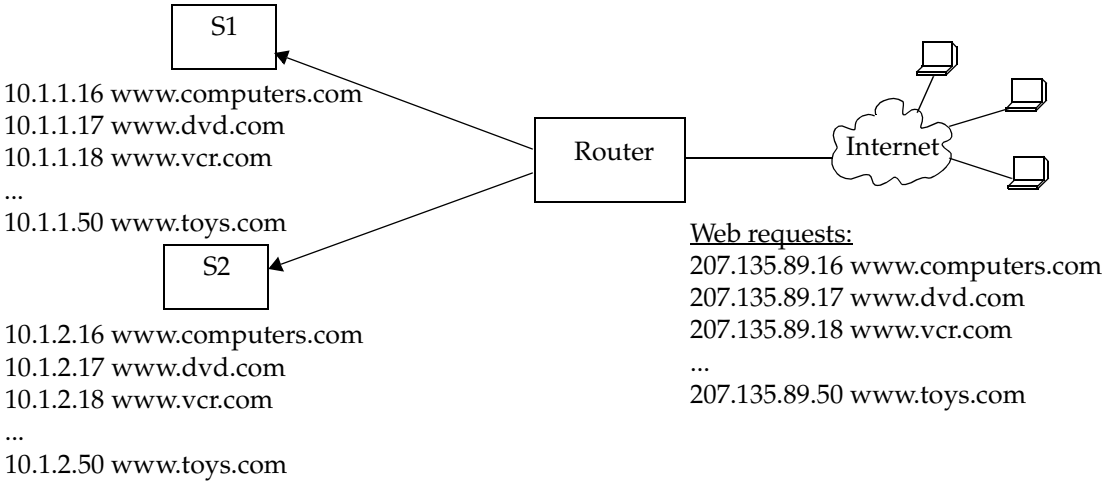
Domain Name	Virtual IP	TCP Port	Real Server IP	TCP Port
www.quick.com	207.135.89.16	80	10.1.1.1	80
ftp.quick.com	207.135.89.16	21	10.1.1.2	21

The network shown above can be created with the following load-balance commands:

```
load-balance create group-name quick-www virtual-ip 207.135.89.16 virtual port 80
protocol tcp
load-balance create group-name quick-ftp virtual-ip 207.135.89.16 virtual port 21
protocol tcp
load-balance add host-to-group 10.1.1.1 group-name quick-www port 80
load-balance add host-to-group 10.1.1.2 group-name quick-ftp port 21
```

Virtual IP Address Ranges

ISPs who provide web hosting services for their clients require a large number of virtual IP addresses (VIPs). The **load-balance create vip-range-name** and **load-balance add host-to-vip-range** commands were created specifically for this. An ISP can create a range of VIPs for up to an entire class C network with the **load-balance create vip-range-name** command. Once the vip-range is in place, the ISP can then create the corresponding secondary addresses on their destination servers. Once these addresses have been created, the ISP can add these servers to the vip-range with the **load-balance add host-to-vip-range** command. These two commands combined help ISPs take advantage of web servers like Apache which serve different web pages based on the destination address in the http request. The following example illustrates this:.



Group Name	Virtual IP	TCP Port	Destination Server IP	TCP Port
www.computers.com	207.135.89.16	80	S1: 10.1.1.16 S2: 10.1.2.16	80
www.dvd.com	207.135.89.17	80	S1: 10.1.1.17 S2: 10.1.2.17	80
www.vcr.com	207.135.89.18	80	S1: 10.1.1.18 S2: 10.1.2.18	80
www.toys.com	207.135.89.50	80	S1: 10.1.1.50 S2: 10.1.2.50	80

The network shown in the previous example can be created with the following load-balance commands:

```
load-balance create vip-range-name mywwwrange 207.135.89.16-207.135.89.50
virtual-port 80 protocol tcp
load-balance add host-to-vip-range 10.1.1.16-10.1.1.50 vip-range-name mywwwrange
port 80
load-balance add host-to-vip-range 10.1.2.16-10.1.2.50 vip-range-name mywwwrange
port 80
```

Web Caching

Web caching provides a way to store frequently accessed Web objects on a cache of local servers. Each HTTP request is transparently redirected by the GSR to a configured cache server. When a user first accesses a Web object, that object is stored on a cache server. Each subsequent request for the object uses this cached object. Web caching allows multiple users to access Web objects stored on local servers with a much faster response time than accessing the same objects over a WAN connection. This can also result in substantial cost savings by reducing the WAN bandwidth usage.

Note: The GSR itself does not act as cache for web objects. It redirects HTTP requests to local servers on which the web objects are cached. One or more local servers are needed to work as cache servers with the GSR's web caching function.

Configuring Web Caching

The following are the steps in configuring Web caching on the GSR:

1. Create the cache group (a list of cache servers) to cache Web objects.
2. Specify the hosts whose HTTP requests will be redirected to the cache servers. This step is optional; if you do not explicitly define these hosts, then *all* HTTP requests are redirected.
3. Apply the caching policy to an outbound interface to redirect HTTP traffic on that interface to the cache servers.

Creating the Cache Group

You can specify either a range of IP addresses or a list of up to four IP addresses to define the servers when the cache group is created. If you specify multiple servers, load balancing is based on the destination address of the request. If any cache server fails, traffic is redirected to the other active servers.

To create the cache group, enter the following command in Configure mode:

Create the cache group.	**web-cache** *<cache-name>* **create server-list** *<server-list-name>* **range** *<ipaddr-range>***|list** *<ipaddr-list>*

Specifying the Client(s) for the Cache Group (Optional)

You can explicitly specify the hosts whose HTTP requests are or are not redirected to the cache servers. If you do not explicitly specify these hosts, then *all* HTTP requests are redirected to the cache servers.

To specify the clients or non-clients for the cache group, enter the following commands in Configure mode:

Define hosts whose requests are redirected to cache servers.	**web-cache** *<cache-name>* **permit hosts range** *<ipaddr-range>***|list** *<ipaddr-list>* | **acl** *<acl-name>*
Define hosts whose requests are *not* redirected to cache servers.	**web-cache** *<cache-name>* **deny hosts range** *<ipaddr-range>***|list** *<ipaddr-list>* | **acl** *<acl-name>*

Redirecting HTTP Traffic on an Interface

To start the redirection of HTTP requests to the cache servers, you need to apply a caching policy to a specific outbound interface. This interface is typically an interface that connects to the Internet.

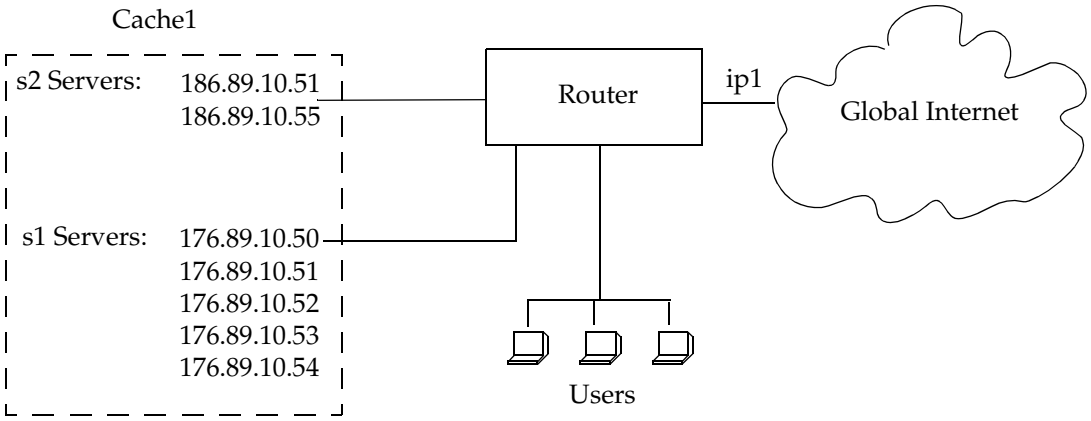
Note: By default, the GSR redirects HTTP requests on port 80. Secure HTTP (https) requests do not run on port 80, therefore these types of requests are not redirected by the GSR.

To redirect outbound HTTP traffic to the cache servers, enter the following command in Configure mode:

Apply caching policy to outbound interface.	**web-cache** *<cache-name>* **apply interface** *<interface-name>*

Configuration Example

In the following example, a cache group of seven local servers is configured to store Web objects for users in the local network:



The following commands configure the cache group 'cache1' that contains the servers shown in the figure above and applies the caching policy to the interface 'ip1':

```
gs/r(config)# web-cache cache1 create server-list s1 range
   "176.89.10.50 176.89.10.54"
gs/r(config)# web-cache cache1 create server-list s2 list "186.89.10.51
186.89.10.55"
gs/r(config)# web-cache cache1 apply interface ip1
```

Note that in this example, HTTP requests from *all* hosts in the network are redirected as there are no **web-cache permit** or **web-cache deny** commands.

Other Configurations

This section discusses other commands that may be useful in configuring Web caching in your network.

Bypassing Cache Servers

Some Web sites require source IP address authentication for user access, therefore HTTP requests for these sites *cannot* be redirected to the cache servers. To specify the sites for which HTTP requests are not redirected to the cache servers, enter the following command in Configure mode:

Define destination sites to which HTTP requests are sent directly.	**web-cache** *<cache-name>* **create bypass-list range** *<ipaddr-range>* | **list** *<ipaddr-list>* | **acl** *<acl-name>*

Proxy Server Redundancy

Some networks use proxy servers that receive HTTP requests on a non-standard port number (i.e., not port 80). When the proxy server is available, all HTTP requests are handled by the proxy server. The GSR can provide proxy server redundancy by transparently redirecting HTTP connections to the cache servers should the proxy server fail. To achieve this, the GSR must be configured to redirect HTTP requests on the (non-standard) HTTP port used by the proxy server.

To redirect HTTP requests to a non-standard HTTP port number, enter the following command in Configure mode:

Specify non-standard HTTP port.	**web-cache** *<cache-name>* **set http-port** *<port number>*

Distributing Frequently-Accessed Sites Across Cache Servers

The GSR uses the destination IP address of the HTTP request to determine which cache server to send the request. However, if there is a Web site that is being accessed very frequently, the cache server serving requests for this destination address may become overloaded with user requests. You can specify that certain destination addresses be distributed across the cache servers in a round-robin manner.

To distribute a specified destination address across cache servers, enter the following command in Configure mode:

Distribute destination address across cache servers.	**web-cache** *<cache-name>* **set round-robin range** *<ipaddr-range>* | **list** *<ipaddr-list>*

Monitoring Web-Caching

To display Web-caching information, enter the following commands in Enable mode.

Show information for all caching policies and all server lists.	**web-cache show all**
Show caching policy information.	**web-cache show cache-name** <*cache-name*> **|all**
Show cache server information.	**web-cache show servers cache** <*cache-name*> **|all**

Chapter 16

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 GSR 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 GSR uses IPX RIP to create and maintain a database of internetwork routing information. The GSR'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.

GSR'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. Through 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 GSR uses IPX SAP to create and maintain a database of internetwork service information. The GSR’s implementation of SAP allows the following exchanges of information:

- Workstations locate the name and address of the nearest server of certain type
- Routers request the names and addresses of either all or certain type of servers
- Servers respond to the workstation’s or router’s request

- Routers make periodic broadcasts to make sure all other routers are aware of the internetwork configuration
- Routers perform broadcasting whenever they detect a change in the internetwork configurations

Configuring IPX RIP & SAP

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

IPX RIP

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

IPX SAP

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

Creating IPX Interfaces

When you create IPX interfaces on the GSR, 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.

Note: You cannot assign IPX interfaces for LAN and WAN to the same VLAN. In order for these two types of IPX interfaces to coexist on the GSR, each type must be assigned to different VLANs.

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

Configuring IPX Routing

By default, IPX routing is enabled on the GSR.

Enabling IPX RIP

IPX RIP is enabled by default on the GSR. 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 GSR. 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 GSR 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, 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 GSR 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** <*SrcNetwork Node*> <*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** <*Interface Name*> **input|output [logging [on|off]]**

Creating an IPX Type 20 Access Control List

IPX type 20 access control lists control the forwarding of IPX type 20 packets. To create an IPX type 20 access control list, enter the following command in Configure mode:

Create an IPX type 20 access control list.	**acl** *<name>* **permit|deny ipxtype20**

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, enter the following command in 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 GNS Access Control List

IPX GNS access control lists control which SAP services the GSR can reply with to a get nearest server (GNS) request. To create an IPX GNS access control list, enter the following command in Configure mode:

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

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

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

Monitoring an IPX Network

The GSR 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> <ServiceName> <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**
Show IPX RIP/SAP table summary.	**ipx show tables summary**

Configuration Examples

This example performs the following configuration:

- Creates IPX interfaces
- Adds static RIP routes
- Adds static SAP entries
- Adds a RIP access list
- Adds a SAP access list
- Adds a GNS 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
!
!IPX type 20 access list
acl 300 deny ipxtype20
!
!IPX type 20 inbound filter to interface ipx2
acl 300 apply interface ipx2 input
!
!GNS Access List
acl 300 deny ipxgns A.01:03:05:07:02:03 0004 FILESERVER2
acl 200 apply interface ipx2 output
```

Chapter 17

Access Control List Configuration Guide

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

This chapter explains how to configure and use Access Control Lists (ACLs) on the GSR. ACLs are lists of selection criteria for specific types of packets. When used in conjunction with certain GSR functions, ACLs allow you to restrict Layer-3/4 traffic going through the router.

This chapter contains the following sections:

- "ACL Basics" on page 236 explains how ACLs are defined and how the GSR evaluates them.
- "Creating and Modifying ACLs" on page 240 describes how to edit ACLs, either remotely or by using the GSR's built-in ACL Editor function.
- "Using ACLs" on page 242 describes the different kinds of ACLs: Interface ACLs, Service ACLs, and Profile ACLs, and gives examples of their usage.
- "Enabling ACL Logging" on page 249 explains how to log information about packets that are permitted or denied because of an ACL.
- "Monitoring ACLs" on page 250 lists the commands you can use to display information about ACLs active on the GSR.

ACL Basics

An ACL consists of one or more *rules* describing a particular type of IP or IPX traffic. ACLs can be simple, consisting of only one rule, or complicated with many rules. Each rule tells the GSR to either permit or deny packets that match selection criteria specified in the 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*.

For example, the following ACL has a rule that permits all IP packets from subnet 10.2.0.0/16 to go through the GSR:

```
acl 101 permit ip 10.2.0.0/16
```

Defining Selection Criteria in ACL Rules

Selection criteria in the rule describe characteristics about a packet. In the example above, the selection criteria are IP packets from 10.2.0.0/16.

The selection criteria you can specify in an ACL rule depends on the type of ACL you are creating. For IP, TCP, and UDP ACLs, the following selection criteria can be specified:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Type of Service (TOS)

For IPX ACLs, the following selection criteria can be specified:

- Source network address
- Destination network address
- Source IPX socket
- Destination IPX socket

These selection criteria are specified as *fields* of an ACL rule. The following syntax description shows the fields of an IP ACL rule:

```
acl <name> permit|deny ip <SrcAddr/Mask> <DstAddr/Mask> <SrcPort> <DstPort> <tos>
```

Note: The **acl permit|deny ip** command restricts traffic for all IP-based protocols, such as TCP, UDP, ICMP, and IGMP. Variants of the a**cl permit|deny ip** command exist that allow you to restrict traffic for a specific IP-based protocol; for example, the **acl permit|deny tcp command** lets you restrict only TCP traffic. These variants have the same syntax and fields as the **acl permit|deny ip** command.

The following syntax description shows the fields of an IPX ACL rule:

```
acl <name> permit|deny ipx <SrcAddr> <SrcSocket> <DstAddr> <DstSocket>
<SrcNetMask> <DstNetMask>
```

Each field in an ACL rule is position sensitive. For example, for a rule for TCP traffic, the source address must be followed by the destination address, followed by the source socket and the destination socket, and so on.

Not all fields of an ACL 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, a rule for TCP can use socket port numbers, while a rule for IPX can use a network node address.

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, use the keyword **any**. For example, the following ACL rule denies SMTP traffic between any two hosts:

```
acl nosmtp deny tcp any any smtp smtp
```

Note that in the above example, the <*tos*> (Type of Service) field is not specified and is treated as a wildcard. The **any** keyword is needed only to skip a wildcard field in order to explicitly specify another field that is further down in the rule. If there are no other fields to specify, the **any** keyword is not necessary. For example, the following ACL permits all IP traffic to go through:

```
acl yesip permit ip
```

How ACL Rules are Evaluated

For an ACL with multiple rules, the ordering of the rules is important. When the GSR checks a packet against an ACL, it goes through each rule in the ACL sequentially. If a packet matches a rule, it is forwarded or dropped based on the **permit** or **deny** keyword in the rule. All subsequent rules are ignored. That is, a first-match algorithm is used. There is no hidden or implied ordering of ACL rules, nor is there precedence attached to each field. The GSR simply goes down the list, one rule at a time, until there is a match. Consequently, rules that are more specific (that is, with more selection criteria) 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 does 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
```

all TCP packets would 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 would match the first rule and be allowed to go through. The second rule would not be looked at since the first match determines the action taken 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 this rule.

This is done for security reasons. If an ACL is misconfigured, and a packet that should be allowed to go through is blocked because of the implicit deny rule, the worst 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. Thus, the implicit deny rule serves as a line of defense against accidental misconfiguration of ACLs.

To illustrate how the implicit deny rule is used, consider 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 is dropped. This is because the third rule (the implicit deny rule) matches all packets.

Although the implicit deny rule may seem 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, you 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 would not match the first rule, but would match the implicit deny rule. As a result, no packets would be allowed to go through. The first rule is simply a subset of the second rule. To allow packets from subnets other than 10.1.20.0/24 to go through, you would have to explicitly define a rule to permit other packets to go through.

To correct the above example and let packets from other subnets enter the GSR, you 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 forwards all packets that are not denied by the first rule.

Because of the implicit deny rule, an ACL works similarly to a firewall that is elected to deny all traffic. You create ACL rules that punch "holes" into the firewall to permit specific types of traffic; for example, traffic from a specific subnet or traffic from a specific application.

Allowing External Responses to Established TCP Connections

Typically organizations that are connected to the outside world implement ACLs to deny access to the internal network. If an internal user wishes to connect to the outside world, the request is sent; however any incoming replies may be denied because ACLs prevent them from going through. To allow external responses to internally generated requests, you would have to create an ACL to allow responses from each specific outside host. If the number of outside hosts that internal users need to access is large or changes frequently, this can be difficult to maintain.

To address this problem, the GSR can be configured to accept outside TCP responses into the internal network, provided that the TCP connection was initiated internally. Otherwise, it will be rejected. To do this, enter the following command in Configure Mode:

Allow TCP responses from external hosts, provided the connection was established internally.	**acl** <*name*> **permit tcp established**

The following ACL illustrates this feature:

```
acl 101 permit tcp established
acl 101 apply interface int1 input
```

Any incoming TCP packet on interface int1 is examined, and if the packet is in response to an internal request, it is permitted; otherwise, it is rejected. Note that the ACL contains no restriction for outgoing packets on interface int1, since internal hosts are allowed to access the outside world.

Creating and Modifying ACLs

The GSR provides two mechanisms for creating and modifying ACLs:

- Editing ACLs on a remote host and uploading them to the GSR using TFTP or RCP
- Using the GSR's ACL Editor

The following sections describe these methods.

Editing ACLs Offline

You can create and edit ACLs on a remote host and then upload them to the GSR with TFTP or RCP. With this method, you use a text editor on a remote host to edit, delete, replace, or reorder ACL rules in a file. Once the changes are made, you can then upload the ACLs to the GSR using TFTP or RCP and make them take effect on the running system. The following example describes how you can use TFTP to help maintain ACLs on the GSR.

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 int12 input
```

The first command, **no acl ***, negates all commands that start with the keyword, "acl". This tells the GSR to remove the application and the definition of any ACL. You can be more selective if you want to remove only ACL commands related to, for instance, ACL 101 by entering, **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 int12.

If the changes are accessible from a TFTP server, you can upload and make the changes take effect by issuing commands like the following:

```
gs/r# copy tftp://10.1.1.12/config/acl.changes to scratchpad
gs/r# copy scratchpad to active
```

The first **copy** command uploads the file acl.changes from a TFTP server and puts the commands into the temporary configuration area, the scratchpad. The administrator can re-examine the changes if necessary before committing the changes to the running system. The second **copy** command makes the changes take effect by copying from the scratchpad to the active running system.

If you need to re-order or modify the ACL rules, you must make the changes in the acl.changes file on the remote host, upload 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 GSR provides a simpler and more user-friendly mechanism to maintain ACLs: the ACL Editor.

The ACL Editor can only be accessed within Configure mode using the **acl-edit** command. You edit an ACL 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 ACL 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 reapply it after changes are made. The process is automatic.

Using ACLs

It is important to understand that an ACL is simply a definition of packet characteristics specified in a set of rules. An ACL must be *enabled* in one of the following ways:

- Applying an ACL to an interface, which permits or denies traffic to or from the GSR. ACLs used in this way are known as *Interface ACLs*.
- Applying an ACL to a service, which permits or denies access to system services provided by the GSR. ACLs used in this way are known as *Service ACLs*.
- Associating an ACL with **ip-policy**, **nat**, **port mirroring**, **rate-limit**, or **web-cache** commands, which specifies the criteria that packets, addresses, or flows must meet in order to be relevant to these GSR features. ACLs used in this way are known as *Profile ACLs*.

These uses of ACLs are described in the following sections.

Applying ACLs to Interfaces

An ACL can be applied to an interface to examine either inbound or outbound traffic. Inbound traffic is traffic coming into the GSR. Outbound traffic is traffic going out of the GSR. 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 traffic, 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 the GSR at an interface where an inbound ACL is applied, the GSR compares the packet to the rules specified by that ACL. If it is permitted, the packet is allowed into the GSR. 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 to 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.

When you apply an ACL to an interface, you can also specify whether the ACL can be modified or removed from the interface by an external agent (such as the Policy Manager application). Note that for an external agent to modify or remove an applied ACL from an interface, the **acl-policy enable external** command must be in the configuration.

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 GSR 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 reasons, whenever possible, you should create and apply an ACL to the inbound interface.

To apply an ACL to an interface, enter the following command in Configure mode:

Apply ACL to an interface.	**acl** <*name*> **apply interface** <*interface name*> **input|output [logging on|off|deny-only|permit-only][policy local|external]**

Applying ACLs to Services

ACLs can also be created to permit or deny access to system services provided by the GSR; for example, HTTP or Telnet servers. 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 you 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 GSR. As a result, only inbound traffic to the GSR is checked. Destination address and port information is ignored; therefore if you are defining a Service ACL, you do not need to specify destination information.

Note: If a service does not have an ACL applied, that service is accessible to everyone. To control access to a service, an ACL must be used.

To apply an ACL to a service, enter the following command in Configure mode:

Apply ACL to a service.	**acl** <*name*> **apply service** <*service name*> **[logging [on|off]]**

Using ACLs as Profiles

You can use the **acl** command to define a *profile*. A profile specifies the criteria that addresses, flows, hosts, or packets must meet to be relevant to certain GSR features. Once you have defined an ACL profile, you can use the profile with the configuration command for that feature. For example, the Network Address Translation (NAT) feature on the GSR allows you to create address pools for dynamic bindings. You use ACL profiles to represent the appropriate pools of IP addresses.

The following GSR features use ACL profiles:

GSR Feature	ACL Profile Usage
IP policy	Specifies the packets that are subject to the IP routing policy.
Dynamic NAT	Defines local address pools for dynamic bindings.
Port mirroring	Defines traffic to be mirrored.
Rate limiting	Specifies the incoming traffic flow to which rate limiting is applied.
Web caching	Specifies which HTTP traffic should always (or never) be redirected to the cache servers. Specifies characteristics of Web objects that should not be cached.

Note the following about using Profile ACLs:

- Only IP ACLs can be used as Profile ACLs. ACLs for non-IP protocols *cannot* be used as Profile ACLs.
- The **permit**/**deny** keywords, while required in the ACL rule definition, are *disregarded* in the configuration commands for the above-mentioned features. In other words, the configuration commands will act upon a specified Profile ACL whether or not the Profile ACL rule contains the **permit** or **deny** keyword.
- Unlike with other kinds of ACLs, there is no implicit deny rule for Profile ACLs.

- Only certain ACL rule parameters are relevant for each configuration command. For example, the configuration command to create NAT address pools for dynamic bindings (the **nat create dynamic** command) only looks at the source IP address in the specified ACL rule. The destination IP address, ports, and TOS parameters, if specified, are ignored.

Specific usage of Profile ACLs is described in more detail in the following sections.

Using Profile ACLs with the IP Policy Facility

The IP policy facility uses a Profile ACL to define criteria that determines which packets should be forwarded according to an IP policy. Packets that meet the criteria defined in the Profile ACL are forwarded according to the **ip-policy** command that references the Profile ACL.

For example, you can define an IP policy that causes all telnet packets travelling from source network 9.1.1.0/24 to destination network 15.1.1.0/24 to be forwarded to destination address 10.10.10.10. You use a Profile ACL to define the selection criteria (in this case, telnet packets travelling from source network 9.1.1.0/24 to destination network 15.1.1.0/24). Then you use an **ip-policy** command to specify what happens to packets that match the selection criteria (in this example, forward them to address 10.10.10.10). The following commands illustrate this example.

This command creates a Profile ACL called *prof1* that uses as its selection criteria all telnet packets travelling from source network 9.1.1.0/24 to destination network 15.1.1.0/24:

```
gs/r(config)# acl prof1 permit ip 9.1.1.0/24 15.1.1.0/24 any any telnet
0
```

This Profile ACL is then used in conjunction with the **ip-policy** command to cause packets matching prof1's selection criteria (that is, telnet packets travelling from 9.1.1.0/24 to 15.1.1.0/24) to be forwarded to 10.10.10.10:

```
gs/r(config)# ip-policy p5 permit profile prof1 next-hop-list
10.10.10.10
```

See Chapter 13, "IP Policy-Based Forwarding Configuration Guide," for more information on using the **ip-policy** command.

Using Profile ACLs with the Traffic Rate Limiting Facility

Traffic rate limiting is a mechanism that allows you to control bandwidth usage of incoming traffic on a per-flow basis. A flow meeting certain criteria can have its packets re-prioritized or dropped if its bandwidth usage exceeds a specified limit.

For example, you can cause packets in flows from source address 1.2.2.2 to be dropped if their bandwidth usage exceeds 10 Mbps. You use a Profile ACL to define the selection criteria (in this case, flows from source address 1.2.2.2). Then you use a **rate-limit** command to specify what happens to packets that match the selection criteria (in this example, drop them if their bandwidth usage exceeds 10 Mbps). The following commands illustrate this example.

This command creates a Profile ACL called *prof2* that uses as its selection criteria all packets originating from source address 1.2.2.2:

```
gs/r(config)# acl prof2 permit ip 1.2.2.2
```

The following command creates a *rate limit definition* that causes flows matching Profile ACL prof2's selection criteria (that is, traffic from 1.2.2.2) to be restricted to 10 Mbps for each flow. If this rate limit is exceeded, the packets are dropped.

```
gs/r(config)# rate-limit client1 input acl prof2 rate-limit 10000000
exceed-action drop-packets
```

When the rate limit definition is applied to an interface (with the **rate-limit apply interface** command), packets in flows originating from source address 1.2.2.2 are dropped if their bandwidth usage exceeds 10 Mbps.

See "Limiting Traffic Rate" on page 272 for more information on using the **rate-limit** command.

Using Profile ACLs with Dynamic NAT

Network Address Translation (NAT) allows you to map an IP address used within one network to a different IP address used within another network. NAT is often used to map addresses used in a private, local intranet to one or more addresses used in the public, global Internet.

The GSR supports two kinds of NAT: *static* NAT and *dynamic* NAT. With dynamic NAT, an IP address within a range of local IP addresses is mapped to an IP address within a range of global IP addresses. For example, you can configure IP addresses on network 10.1.1.0/24 to use an IP address in the range of IP addresses in network 192.50.20.0/24. You can use a Profile ACL to define the ranges of local IP addresses.

The following command creates a Profile ACL called *local*. The local profile specifies as its selection criteria the range of IP addresses in network 10.1.1.0/24..

```
gs/r(config)# acl local permit ip 10.1.1.0/24
```

Note: When a Profile ACL is defined for dynamic NAT, only the source IP address field in the **acl** statement is evaluated. All other fields in the **acl** statement are ignored.

Once you have defined a Profile ACL, you can then use the **nat create dynamic** command to bind the range of IP addresses defined in the local profile to a range in network 192.50.20.0/24.

```
gs/r(config)# nat create dynamic local-acl-pool local global-pool
192.50.20.10/24
```

See Chapter 14, "Network Address Translation Configuration Guide," for more information on using dynamic NAT.

Using Profile ACLs with the Port Mirroring Facility

Port mirroring refers to the GSR's ability to copy traffic on one or more ports to a "mirror" port, where an external analyzer or probe can be attached. In addition to mirroring traffic on one or more ports, the GSR can mirror traffic that matches selection criteria defined in a Profile ACL.

For example, you can mirror all IGMP traffic on the GSR. You use a Profile ACL to define the selection criteria (in this example, all IGMP traffic). Then you use a **port mirroring** command to copy packets that match the selection criteria to a specified mirror port. The following commands illustrate this example.

This command creates a Profile ACL called *prof3* that uses as its selection criteria all IGMP traffic on the GSR:

```
gs/r(config)# acl prof3 permit igmp
```

The following command causes packets matching Profile ACL prof3's selection criteria (that is, all IGMP traffic) to be copied to mirror port et.1.2.

```
gs/r(config)# port mirroring monitor-port et.1.2 target-profile prof3
```

See "Configuring the GSR for Port Mirroring" on page 277 for more information on using the **port mirroring** command.

Using Profile ACLs with the Web Caching Facility

Web caching is the GSR's ability to direct HTTP requests for frequently accessed Web objects to local cache servers, rather than to the Internet. Since the HTTP requests are handled locally, response time is faster than if the Web objects were retrieved from the Internet.

You can use Profile ACLs with Web caching in two ways:

- Specifying which HTTP traffic should always (or never) be redirected to the cache servers
- Specifying characteristics of Web objects that should not be cached

Redirecting HTTP Traffic to Cache Servers

You can use a Profile ACL to specify which HTTP traffic should always (or never) be redirected to the cache servers. (By default, when Web caching is enabled, all HTTP traffic from all hosts is redirected to the cache servers unless you specify otherwise.)

For example, you can specify that packets with a source address of 10.10.10.10 and a destination address of 1.2.3.4 always are sent to the Internet and never to the cache servers. The following commands illustrate this example.

This command creates a Profile ACL called *prof4* that uses as its selection criteria all packets with a source address of 10.10.10.10 and a destination address of 1.2.3.4 :

```
gs/r(config)# acl prof4 permit ip 10.10.10.10 1.2.3.4
```

The following command creates a *Web caching policy* that prevents packets matching Profile ACL prof4's selection criteria (that is, packets with a source address of 10.10.10.10 and a destination address of 1.2.3.4) from being redirected to a cache server. Packets that match the profile's selection criteria are sent to the Internet instead.

```
gs/r(config)# web-cache policy1 deny hosts profile prof4
```

When the Web caching policy is applied to an interface (with the **web-cache apply interface** command), HTTP traffic with a source address of 10.10.10.10 and a destination address of 1.2.3.4 goes to the Internet instead of to the cache servers.

Preventing Web Objects From Being Cached

You can also use a Profile ACL to prevent certain Web objects from being cached. For example, you can specify that information in packets originating from Internet site 1.2.3.4 and destined for local host 10.10.10.10 not be sent to the cache servers. The following commands illustrate this example.

This command creates a Profile ACL called *prof5* that uses as its selection criteria all packets with a source address of 1.2.3.4 and a destination address of 10.10.10.10:

```
gs/r(config)# acl prof5 permit ip 1.2.3.4 10.10.10.10
```

To have packets matching Profile ACL prof5's selection criteria bypass the cache servers, use the following command:

```
gs/r(config)# web-cache policy1 create bypass-list profile prof5
```

When the Web caching policy is applied to an interface, information in packets originating from source address 1.2.3.4 and destined for address 10.10.10.10 is not sent to the cache servers.

See "Web Caching" on page 219 for more information on using the **web-cache** command.

Enabling ACL Logging

To see whether incoming packets are permitted or denied because of an ACL, you 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 GSR, the same information will also be sent to the Syslog server.

Before enabling ACL Logging, you 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, you should consider the potential performance impact before turning on ACL Logging.

Monitoring ACLs

The GSR provides a display of ACL configurations active in the system.

To display ACL information, enter the following commands 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 18

Security Configuration Guide

Security Overview

The GSR provides security features that help control access to the GSR and filter traffic going through the GSR. Access to the GSR can be controlled by:

- Enabling RADIUS
- Enabling TACACS
- Enabling TACACS Plus
- Password authentication

Traffic filtering on the GSR enables:

- Layer-2 security filters - Perform filtering on source or destination MAC addresses.
- Layer-3 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 GSR, for example, Telnet server and HTTP server.

Note: Currently, Source Filtering is available on GSR WAN cards; however, application must take place on the entire WAN card.

Configuring GSR Access Security

This section describes the following methods of controlling access to the GSR:

- RADIUS
- TACACS
- TACACS Plus
- Passwords

Configuring RADIUS

You can secure login or Enable mode access to the GSR by enabling a Remote Authentication Dial-In Service (RADIUS) client. A RADIUS server responds to the GSR RADIUS client to provide authentication.

You can configure up to five RADIUS server targets on the GSR. A timeout is set to tell the GSR how long to wait for a response from RADIUS servers.

To configure RADIUS security, enter the following commands in Configure mode:

Specify a RADIUS server.	**radius set server** *<hostname or IP-addr>*
Set the RADIUS time to wait for a RADIUS server reply.	**radius set timeout** *<number>*
Determine the GSR action if no server responds.	**radius set last-resort password|succeed**
Enable RADIUS.	**radius enable**
Cause RADIUS authentication at user login or when user tries to access Enable mode.	**radius authentication login|enable**
Logs specified types of command to RADIUS server.	**radius accounting command level <***level***>**
Logs to RADIUS server when shell is stopped or started on GSR.	**radius accounting shell start|stop|all**
Logs to RADIUS server SNMP changes to startup or active configuration.	**radius accounting snmp active|startup**
Logs specified type(s) of messages to RADIUS server.	**radius accounting system fatal|error|warning|info**

Monitoring RADIUS

You can monitor RADIUS configuration and statistics within the GSR.

To monitor RADIUS, enter the following commands in Enable mode:

Show RADIUS server statistics.	**radius show stats**
Show all RADIUS parameters.	**radius show all**

Configuring TACACS

In addition, Enable mode access to the GSR can be made secure by enabling a Terminal Access Controller Access Control System (TACACS) client. Without TACACS, TACACS Plus, or RADIUS enabled, only local password authentication is performed on the GSR. The TACACS client provides user name and password authentication for Enable mode. A TACACS server responds to the GSR TACACS client to provide authentication.

You can configure up to five TACACS server targets on the GSR. A timeout is set to tell the GSR 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 server** *<hostname or IP-addr>*
Set the TACACS time to wait for a TACACS server reply.	**tacacs set timeout** *<number>*
Determine GSR 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 GSR.

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 TACACS Plus

You can secure login or Enable mode access to the GSR by enabling a TACACS Plus client. A TACACS Plus server responds to the GSR TACACS Plus client to provide authentication.

You can configure up to five TACACS Plus server targets on the GSR. A timeout is set to tell the GSR how long to wait for a response from TACACS Plus servers.

To configure TACACS Plus security, enter the following commands in Configure mode:

Specify a TACACS Plus server.	**tacacs-plus set server** <*hostname or IP-addr*>
Set the TACACS Plus time to wait for a TACACS Plus server reply.	**tacacs-plus set timeout** <*number*>
Determine the GSR action if no server responds.	**tacacs-plus set last-resort password|succeed**
Enable TACACS Plus.	**tacacs-plus enable**
Cause TACACS Plus authentication at user login or when user tries to access Enable mode.	**tacacs-plus authentication login|enable**
Cause TACACS Plus authentication at user login or when user tries to access Enable mode.	**tacacs-plus authentication login|enable**
Logs specified types of command to TACACS Plus server.	**tacacs-plus accounting command level <***level***>**
Logs to TACACS Plus server when shell is stopped or started on GSR.	**tacacs-plus accounting shell start|stop|all**
Logs to TACACS Plus server SNMP changes to startup or active configuration.	**tacacs-plus accounting snmp active|startup**
Logs specified type(s) of messages to TACACS Plus server.	**tacacs-plus accounting system fatal|error|warning|info**

Monitoring TACACS Plus

You can monitor TACACS Plus configuration and statistics within the GSR.

To monitor TACACS Plus, enter the following commands in Enable mode:

Show TACACS Plus server statistics.	**tacacs-plus show stats**
Show all TACACS Plus parameters.	**tacacs-plus show all**

Configuring Passwords

The GSR provides password authentication for accessing the User and Enable modes. If TACACS is not enabled on the GSR, only local password authentication is performed.

To configure GSR 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>*

Layer-2 Security Filters

Layer-2 security filters on the GSR 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 GSR 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 GSR 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

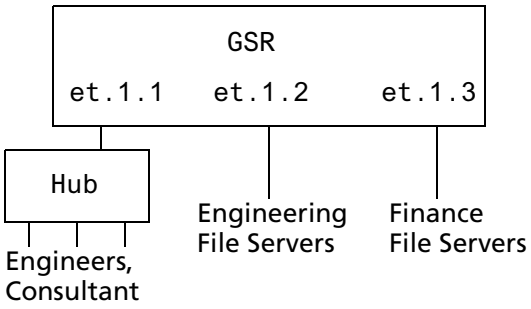


Figure 23. Source Filter Example

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
```

Layer-3 Access Control Lists (ACLs)

Access Control Lists (ACLs) allow you to restrict Layer-3 traffic going through the GSR. Each ACL 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.

For information about defining and using ACLs on the GSR, see Chapter 17, "Access Control List Configuration Guide."

Chapter 19

QoS Configuration Guide

QoS & Layer-2/Layer-3/Layer-4 Flow Overview

The GSR allows network managers to identify traffic and set Quality of Service (QoS) policies without compromising wire speed performance. The GSR 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.

The GSR provides three different features to satisfy QoS requirements:

- *Traffic prioritization* allows network administrators to identify and segregate mission-critical network traffic into different priority queues from non-critical network traffic. Once a packet has been identified, it can be assigned into any one of four priorities in order to ensure delivery. Priority can be allocated based on any combination of Layer-2, Layer-3, or Layer-4 traffic.
- *Type of Service (ToS) rewrite* provides network administrators access to the ToS octet in an IP packet. The ToS octet is designed to provide feedback to the upper layer application. The administrator can "mark" packets using the ToS rewrite feature so that the application (a routing protocol, for example) can handle the packet based on a predefined mechanism.
- *Traffic rate limiting* provides network administrators with tools to manage bandwidth resources. The administrator can create an upper limit for a traffic profile, which is based on Layer-3 or Layer-4 information. Traffic that exceeds the upper limit of the profile can either be dropped or re prioritized into another priority queue.

Within the GSR, 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 and Layer-3 & Layer-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 GSR 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 IP 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.

GSR Queuing Policies

You can use one of two queuing policies on the GSR:

- **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.

You can set the queuing policy on a per-port basis. The default queuing policy is strict priority.

Traffic Prioritization for Layer-2 Flows

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), 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.

The VLAN ID in the QoS configuration must match the VLAN ID assigned to the list of ports to which the QoS policy is applied. In a layer-2 only configuration, each port has only one VLAN ID associated with it and the QoS policy should have the same VLAN ID. When different VLANs are assigned to the same port using different protocol VLANs, the layer-2 QoS policy must match the VLAN ID of the protocol VLAN.

Note: In flow mode, you can also ignore the source MAC address and configure the priority based on the destination MAC address only.

Configuring Layer-2 QoS

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*>

Traffic Prioritization for Layer-3 & Layer-4 Flows

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** <*port list*>**|**<*interface-list*>**|any** <*protocol*>**|any** <*tos-mask*>**|any** <*tos-precedence-rewrite*>**|any** <*tos-rewrite*>**|any**

For example, the following command assigns control priority to any traffic coming from the 10.10.11.0 network:

```
gs/r(config)# qos set ip xyz control 10.10.11.0/24
```

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** <*port list*>**|**<*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 GSR Queueing Policy

The GSR queuing policy is set on a system-wide basis. The GSR default queuing policy is strict priority. To change the queuing policy to weighted-fair queuing on the GSR, enter the following command in Configure mode:

Set queuing policy to weighted-fair.	**qos set queuing-policy weighted-fair port** *<port list>***|all-ports**

If you want to revert the GSR queuing policy from weighted-fair to strict priority (default), enter the following command in Configure mode:

Revert the GSR 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 GSR, you can allocate bandwidth for the queues on the GSR. To allocate bandwidth for each GSR queue, enter the following command in Configure mode:

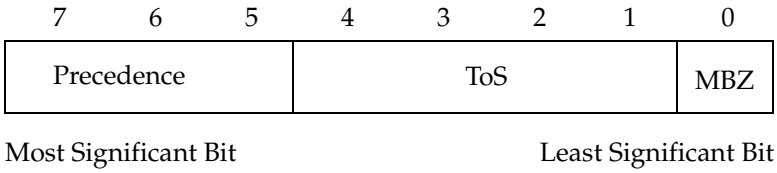
Allocate bandwidth for a weighted-fair queuing policy.	**qos set weighted-fair control** *<percentage>* **high** *<percentage>* **medium** *<percentage>* **low** *<percentage>* **port** *<port list>***|all-ports**

ToS Rewrite

In the Internet, IP packets that use different paths are subject to delays, as there is little inherent knowledge of how to optimize the paths for different packets from different applications or users. The IP protocol actually provides a facility, which has been part of the IP specification since the protocol's inception, for an application or upper-layer protocol to specify how a packet should be handled. This facility is called the Type of Service (ToS) octet.

The ToS octet part of the IP specification, however, has not been widely employed in the past. The IETF is looking into using the ToS octet to help resolve IP quality problems. Some newer routing protocols, like OSPF and IS-IS, are designed to be able to examine the ToS octet and calculate routes based on the type of service.

The ToS octet in the IP datagram header consists of three fields:



- The three-bit Precedence field is used to indicate the priority of the datagram.
- The four-bit ToS field is used to indicate trade-offs between throughput, delay, reliability, and cost.
- The one-bit "must be zero" (MBZ) field is not currently used. (In the GSR configuration, there is no restriction on this bit and it is included as part of the ToS field.)

For example, setting the ToS field to 0010 specifies that a packet will be routed on the most reliable paths. Setting the ToS field to 1000 specifies that a packet will be routed on the paths with the least delay. (Refer to RFC 1349 for the specification of the ToS field value.)

With the ToS rewrite command, you can access the value in the ToS octet (which includes both the Precedence and ToS fields) in each packet. The upper-layer application can then decide how to handle the packet, based on either the Precedence or the ToS field or both fields. For example, you can configure a router to forward packets using different paths, based on the ToS octet. You can also change the path for specific applications and users by changing the Precedence and/or ToS fields.

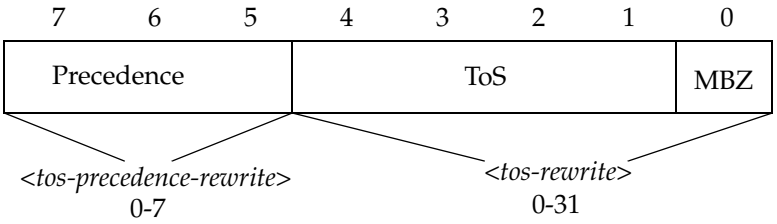
Note: In RFC 2574, the IETF redefined the ToS octet as the "DiffServ" byte. You will still be able to use the ToS rewrite feature to implement DiffServ when this standard is deployed.

Configuring ToS Rewrite for IP Packets

The ToS rewrite for IP packets is set with the **qos set** command in Configure mode. You can define the QoS policy based on any of the following IP fields: source IP address, destination IP address, source port, destination port, ToS, port, or interface.

When an IP packet is received, the ToS field of the packet is ANDed with the <*tos-mask*> and the resulting value is compared with the ANDed value of <*tos*> and <*tos-mask*> of the QoS policy. If the values are equal, the values of the <*tos-rewrite*> and <*tos-precedence-rewrite*> parameters will be written into the packet.

The <*tos*> and <*tos-mask*> parameters use values ranging from 0 to 255. They are used in conjunction with each other to define which bit in the <*tos*> field of the packet is significant. The <*tos-precedence-rewrite*> value ranges from 0 to 7 and is the value that is rewritten in the ToS Precedence field (the first three bits of the ToS octet). The <*tos-rewrite*> value ranges from 0 to 31 and is the value that is rewritten in the ToS field (the last five bits of the ToS octet, which includes both the ToS field and the MBZ bit).



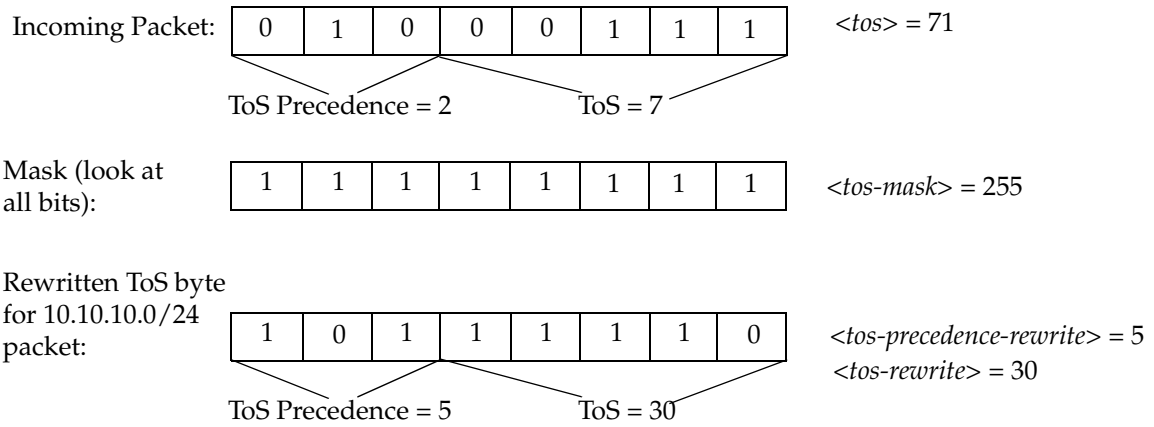
The ToS byte rewrite is part of the QoS priority classifier group. The entire ToS byte can be rewritten or only the precedence part of the ToS byte can be rewritten. If you specify a value for <*tos-precedence-rewrite*>, then only the upper three bits of the ToS byte are changed. If you set <*tos-precedence-rewrite*> to **any** and specify a value for <*tos-rewrite*>, then the upper three bits remain unchanged and the lower five bits are rewritten. If you specify values for both <*tos-precedence-rewrite*> and <*tos-rewrite*>, then the upper three bits are rewritten to the <*tos-precedence-rewrite*> value and the lower five bits are rewritten to the <*tos-rewrite*> value.

For example, the following command will rewrite the ToS Precedence field to 7 if the ToS Precedence field of the incoming packet is 6:

```
gs/r(config)# qos set ip tosp6to7 low any any any any 222 any any 224 7
```

In the above example, the <*tos*> value of 222 (binary value 1101 1110) and the <*tos-mask*> value of 224 (binary value 1110 0000) are ANDed together to specify the ToS Precedence field value of 6 (binary value 110). Changing the value in the <*tos-mask*> parameter determines the bit in the ToS octet field that will be examined.

The following example will rewrite the ToS Precedence and the ToS fields to 5 and 30 if the incoming packet is from the 10.10.10.0/24 network with the ToS Precedence field set to 2 and the ToS field set to 7. (In this example, the MBZ bit is included in the ToS field.) The figure below shows how the parameter values are derived.



The <*tos-mask*> value determines the ToS bit to be examined, which is all eight bits in this example. The following command configures the ToS rewrite for the example:

```
gs/r(config)# qos set ip tos30to7 low 10.10.10.0/24 any any any 71 any
   any 255 5 30
```

Monitoring QoS

The GSR provides display of QoS statistics and configurations contained in the GSR.

To display QoS information, enter the following commands in Enable mode:

Show all IP QoS flows.	**qos show ip**
Show all IPX QoS flows.	**qos show ipx**
Show all Layer-2 QoS flows.	**qos show l2 all-destination all-flow ports** <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

Limiting Traffic Rate

Note: Some commands in this facility require updated GSR hardware. Please refer to the Release Notes for details.

Traffic rate limiting provides the ability to control the usage of a fundamental network resource, bandwidth. It allows you to limit the rate of traffic that flows through the specified interfaces, thus reserving bandwidth for critical applications. Unlike traffic prioritization, traffic rate limiting is a mechanism to control bandwidth usage of incoming traffic on a per flow basis.

A *traffic profile* is used to define the traffic characteristics before an upper limit is assigned. The traffic profile is created using one or more ACLs which can utilize any combination of the parameters supported in the IP ACL. A *rate limiting profile* can then be defined by using the ACL and traffic rate limitations. A single rate limiting profile can have multiple ACLs to define different traffic profiles and traffic rate limitations. When there are multiple traffic profiles, a sequence number is used to identify the order in which the profiles are applied. You can define the action taken on the traffic that exceeds the upper limit: either drop the packets or reset the priority of the traffic. The rate limiting profile is then applied to a logical IP interface.

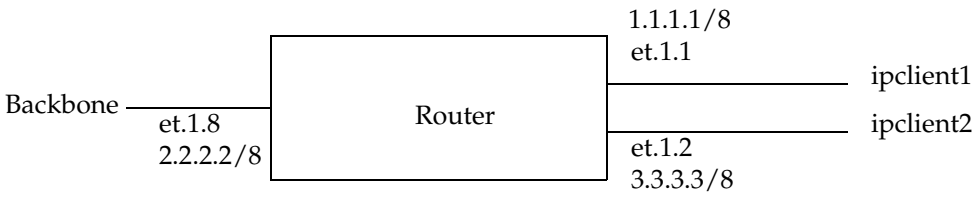
To define a rate limit profile and apply the profile to an interface, enter the following commands in Configure mode:

Define a rate limit profile.	**rate-limit** <*name*> **input acl** <*acl list*> **rate** <*rate-limit*> **exceed-action drop-packets|set-priority-low|set-priority-medium|set-priority-high [sequence** <*number*>**]**
Apply a rate limit profile to an interface.	**rate-limit** <*name*> **apply interface** <*interface*>**|all**

Note: You cannot use non-IP ACLs for rate limit profiles.

Example Configuration

The following is an example of configuring rate limiting on the GSR.



Traffic from two interfaces, 'ipclient1' with IP address 1.2.2.2 and 'ipclient2' with IP address 3.1.1.1, is restricted to 10 Mbps for each flow with the following configuration:

```
vlan create client1 ip
vlan create backbone ip
vlan create client2 ip
vlan add ports et.1.1 to client1
vlan add ports et.1.2 to client2
vlan add ports et.1.8 to backbone
interface create ip ipclient1 vlan client1 address-netmask 1.1.1.1/8
interface create ip ipclient2 vlan client2 address-netmask 3.3.3.3/8
interface create ip backbone vlan backbone address-netmask 2.2.2.2/8
acl 100 permit ip 1.2.2.2
acl 200 permit ip 3.1.1.1
rate-limit client1 input acl 100 rate-limit 10000000 exceed-action drop-packets
rate-limit client2 input acl 200 rate-limit 10000000 exceed-action drop-packets
rate-limit client1 apply interface ipclient1
rate-limit client2 apply interface ipclient2
```

Displaying Rate Limit Information

To show information about rate limit policies, enter the following command in Enable mode:

Show rate limit policy information.	**rate-limit show all| policy-name** <*name*> **| interface** <*interface*>

Chapter 20

Performance Monitoring Guide

Performance Monitoring Overview

The GSR is a full wire-speed layer-2, 3 and 4 switching router. As packets enter the GSR, 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 GSR provides the capability to monitor performance at Layer 2, 3, and 4. Layer-2 performance information is accessible to SNMP through MIB-II and can be displayed by using the **l2-tables** command in the CLI. Layer-3 and 4 performance statistics are accessible to SNMP through RMON/RMON2 and can be displayed by using the **statistics show** command in the CLI. In addition to the monitoring commands listed, you can find more monitoring commands listed in each chapter of the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.

To access statistics on the GSR, enter the following commands in Enable mode:

Show DVMRP routes.	**dvmrp show routes**
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 etherStats statistics.	**statistics show rmon**
Show traffic summary statistics.	**statistics show summary-stats**
Show most active tasks.	**statistics show top**
Show TCP statistics.	**statistics show tcp**
Show UDP statistics.	**statistics show udp**
Show TACACS server statistics.	**tacacs show stats**
Show broadcast monitoring information for ports.	**port show bmon [config][detail][port** *<port list>*]**[stats]**
Show all VLANs.	**vlan list**

Configuring the GSR for Port Mirroring

The GSR allows you to monitor activity with port mirroring. Port mirroring allows you to monitor the performance and activities of one or more ports on the GSR or for traffic defined by an ACL through just a single, separate port. While in Configure mode, you can configure your GSR for port mirroring with a simple command line like the following:

Configure Port Mirroring.	**port mirroring monitor-port** <*port number*> **target-port** <*port list*>|**target-profile** <*acl name*>

Note:
- Port mirroring is available for WAN ports. However, you cannot configure port mirroring on a port-by-port basis. (You can only configure port mirroring for the entire WAN card).
- Only IP ACLs can be specified for port mirroring.

Monitoring Broadcast Traffic

The GSR allows you to monitor broadcast traffic for one or more ports. You can specify that a port be shut down if its broadcast traffic reaches a certain rate limit for a particular period of time. You can also specify the duration of the port shut down. To specify the monitoring of broadcast traffic and the shut down threshold for one or more ports, enter the following command in Configure mode:

Configure monitoring of broadcast traffic.	**port bmon** <*port list*> **rate** <*number*> **duration** <*number*> **shutdown** <*number*>

Chapter 21

RMON Configuration Guide

RMON Overview

You can employ Remote Network Monitoring (RMON) in your network to help monitor traffic at remote points on the network. With RMON, data collection and processing is done with a remote *probe*, namely the GSR. The GSR also includes RMON *agent* software that communicates with a network management station via SNMP. Because information is only transmitted from the GSR to the management station when required, SNMP traffic on the network and the management station's processing load are reduced.

The GSR provides support for both RMON 1 and RMON 2 MIBs, as specified in RFCs 1757 and 2021, respectively. While non-RMON SNMP products allow the monitoring and control of specific network *devices*, RMON 1 returns statistics on network *segments* at the MAC layer. RMON 2 collects statistics on network and application layer *traffic* to show host-to-host connections and the applications and protocols being used. For example, the RMON 2 network layer matrix MIB group can show protocol-specific traffic between pairs of systems which can help to diagnose protocol problems. Note that RMON 2 is not a superset of RMON 1; on the GSR, you can configure both RMON 1 and RMON 2 statistics collection.

Configuring and Enabling RMON

By default, RMON is disabled on the GSR. To configure and enable RMON on the GSR, follow these steps:

1. Turn on the Lite, Standard, or Professional RMON groups by entering the **rmon set lite | standard | professional** command. You can also configure default control tables for the Lite, Standard, or Professional RMON groups by including the **default-tables yes** parameter.
2. Enable RMON on specified ports with the **rmon set ports** command.
3. Optionally, you can configure control tables for the Lite, Standard, or Professional RMON groups. For example, if you chose *not* to create default control tables for the Lite, Standard, or Professional groups, you can configure control table entries for specific ports on the GSR.
4. Use the **rmon enable** command to enable RMON on the GSR.

Example of RMON Configuration Commands

The following are examples of the commands to configure and enable RMON on the GSR:

```
gs/r(config)# show
Running system configuration:
     !
     ! Last modified from Telnet (10.50.89.88) on 1999-04-05 16:52:28
     !
 1 : port flow-bridging et.5.(3-8) *
     !
 2 : interface add ip en0 address-netmask 10.50.6.9/16
     !
 3 : system set contact "usama"
 4 : system set location DIGITAL Equipment Corporation
 5 : system set name "gs/r"
      !
 6 : rmon set ports all-ports
 7 : rmon set lite default-tables yes
 8 : rmon set standard default-tables yes
     !
     ! Set RMON Pro Group with Default Tables ON, cap memory at 4 meg
     ! Pro: protocolDir, protocolDist, addressMap, al/nl-Matrix, al/nl-Host,
     ! al/nl-matrixTopN. userHistory, probeConfig.
     ! Default Tables: one control row per dataSource for protocolDist,
     ! addressMap, al/nl-Host, al/nl-Matrix.
     !
 9 : rmon set professional default-tables yes
10 : rmon set memory 4
11 : rmon enable
```

* To collect layer 2 matrix information, port must be configured for flow-bridging mode. By default, ports on the GSR operate in address-bridging mode.

The next sections describe Lite, Standard, and Professional RMON groups and control tables.

RMON Groups

The RMON MIB groups are defined in RFCs 1757 (RMON 1) and 2021 (RMON 2). On the GSR, you can configure one or more levels of RMON support for a set of ports. Each level—Lite, Standard, or Professional—enables different sets of RMON groups (described later in this section). You need to configure at least one level before you can enable RMON on the GSR.

To specify the support level for RMON groups, use the following CLI command line in Configure mode:

Specifies Lite, Standard, or Professional RMON groups.	**rmon set lite|standard|professional default-tables yes|no**

To specify the ports on which RMON is to be enabled, use the following CLI command line in Configure mode:

Specifies the ports on which RMON is enabled.	**rmon set ports** *<port list>***|allports**

You can configure each level of RMON support independently of each other with default tables on or off. For example, you can configure Lite with default tables on for ports et.1.(1-8) and then configure Standard with no default tables for the same ports. You cannot configure Lite on one set of ports and Standard on another set of ports.

Lite RMON Groups

This section describes the RMON groups that are enabled when you specify the Lite support level. The Lite RMON groups are shown in the table below.

Table 6. Lite RMON Groups

Group	Function
EtherStats	Records Ethernet statistics (for example, packets dropped, packets sent, etc.) for specified ports.
Event	Controls event generation and the resulting action (writing a log entry or sending an SNMP trap to the network management station).
Alarm	Generates an event when specified alarm conditions are met.
History	Records statistical samples for specified ports.

Standard RMON Groups

This section describes the RMON groups that are enabled when you specify the Standard support level. The Standard RMON groups are shown in the table below.

Table 7. Standard RMON Groups

Group	Function
Host	Records statistics about the hosts discovered on the network.
Host Top N	Gathers the top n hosts, based on a specified rate-based statistic. This group requires the hosts group.
Matrix	Records statistics for source and destination address pairs.
Filter	Specifies the type of packets to be matched and how and where the filtered packets should flow (the channel).
Packet Capture	Specifies the capture of filtered packets for a particular channel.

Professional RMON Groups

The Professional RMON groups correspond to the RMON 2 groups defined in RFC 2021. While RMON 1 groups allow for the monitoring of packets at the MAC layer, RMON 2 groups focus on monitoring traffic at the network and application layers.

The Professional RMON groups are shown in the table below.

Table 8. Professional RMON Groups

Group	Function
Protocol Directory	Contains a list of protocols supported by the GSR and monitored by RMON. See the RMON 2 Protocol Directory appendix in the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.
Protocol Distribution	Records the packets and octets for specified ports on a per protocol basis.
Application Layer Host	Monitors traffic at the application layer for protocols defined in the protocol directory.
Network Layer Host	Monitors traffic at the network layer for protocols defined in the Protocol Directory.
Application Layer Matrix (and Top N)	Monitors traffic at the application layer for protocols defined in the Protocol Directory. Top N gathers the top n application layer matrix entries.
Network Layer Matrix (and Top N)	Monitors traffic at the network layer for protocols defined in the Protocol Directory. Top N gathers the top n network layer matrix entries.
Address Map	Records MAC address to network address bindings discovered for specified ports.
User History	Records historical data on user-defined statistics.

Control Tables

Many RMON groups contain both control and data tables. Control tables specify what statistics are to be collected. For example, you can specify the port for which statistics are to be collected and the owner (name, phone, or IP address) for that port. You can change many of the entries in a control table with **rmon** commands. Data tables contain the collected statistics. You cannot change any of the entries in a data table; you can only view the data.

When you specify the Lite, Standard, or Professional RMON groups, you have the option of creating default control tables. A default control table creates a control table entry for every port on the GSR. Creating default control tables essentially configures data collection for every port on the GSR for certain RMON groups. If you do not want this, you can choose not to create the default control tables and then configure the appropriate control tables for the data you wish to collect. Even if you use the default control tables, you can always use the **rmon** commands to modify control table entries.

If you choose to create default control tables, entries are created in the control tables for each port on the GSR for the following groups:

Lite groups:	Etherstats History
Standard groups:	Host Matrix
Professional groups:	Protocol Distribution Address Map Application Layer/Network Layer Host Application Layer/Network Layer Matrix

A row in the control table is created for each port on the GSR, with the owner set to "monitor". If you want, you can change the owner by using the appropriate **rmon** command. See the section "Configuring RMON Groups" in this chapter for more the command to configure a specific group.

Note: Control tables other than the default control tables must be configured with CLI commands, as described in "Configuring RMON Groups".

Using RMON

RMON on the GSR allows you to analyze network traffic patterns, set up alarms to detect potential problems before they turn into real congestive situations, identify heavy network users to assess their possible candidacy for moves to dedicated or higher speed ports, and analyze traffic patterns to facilitate more long-term network planning.

RMON 1 provides layer 2 information. Traffic flowing through the GSR's layer 2 ASIC is collected by RMON 1 groups. RMON 2 in the GSR provides layer 3 traffic information for IP and IPX protocols. Traffic flowing through the GSR's layer 3 ASIC is collected by RMON 2 groups. The GSR's RMON 2 protocol directory contains over 500 protocols that can be decoded for UDP and TCP ports. You can use RMON to see the kinds of protocol traffic being received on a given port.

For example, use the **rmon show protocol-distribution** command to see the kinds of traffic received on a given port:

```
gs/r# rmon show protocol-distribution et.5.5
RMON II Protocol Distribution Table

Index: 506, Port: et.1.7, Owner: monitor
      Pkts     Octets   Protocol
      ----     ------   --------
        19       1586   ether2
        19       1586   ether2.ip-v4
        19       1586   *ether2.ip-v4
         2        192   *ether2.ip-v4.icmp
        17       1394   *ether2.ip-v4.tcp
        17       1394   *ether2.ip-v4.tcp.www-http
```

In the example output above, only HTTP and ICMP traffic is being received on this port. To find out which host or user is using these applications/protocols on this port, use the following command:

```
gs/r# rmon show al-matrix et.5.5
RMON II Application Layer Host Table

Index: 500, Port: et.5.5, Inserts: 4, Deletes: 0, Owner: monitor
SrcAddr         DstAddr            Packets     Octets  Protocol
-------         -------            -------     ------  --------
10.50.89.88     15.15.15.3            1771     272562  *ether2.ip-v4
10.50.89.88     15.15.15.3            1125     211192  *ether2.ip-v4.tcp
10.50.89.88     15.15.15.3            1122     210967  *ether2.ip-v4.tcp.telnet
10.50.89.88     15.15.15.3               3        225  *ether2.ip-v4.tcp.www-http
```

Configuring RMON Groups

As mentioned previously, control tables in many RMON groups specify the data that is to be collected for the particular RMON group. If the information you want to collect is in the default control tables, then you only need to turn on the default tables when you specify the RMON groups (Lite, Standard, or Professional); you do not need to configure entries in the default tables.

The following table shows the **rmon** command that you use to configure each RMON group:

To configure the Address Map group.	**rmon address-map index** <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable	disable**]
To configure the Application Layer Matrix top n entries.	**rmon al-matrix-top-n index** <*index-number*> **matrix-index** <*number*> **ratebase terminal-packets	terminal-octets	all-packets	all-octets duration** <*number*> **size** <*number*> [**owner** <*string*>] [**status enable	disable**]
To configure the Alarm group.	**rmon alarm index** <*index-number*> **variable** <*string*> [**interval** <*seconds*>] [**falling-event-index** <*num*>] [**falling-threshold** <*num*>] [**owner** <*string*>] [**rising-event-index** <*num*>] [**rising-threshold** <*num*>] [**startup rising	falling	both**] [**status enable	disable**] [**type absolute-value	delta-value**]
To configure the Packet Capture group.	**rmon capture index** <*index-number*> **channel-index**<*number*> [**full-action lock	wrap**] [**slice-size** <*number*>] [**download-slice-size** <*number*>] [**download-offset** <*number*>] [**max-octets** <*number*>] [**owner** <*string*>] [**status enable	disable**]
To configure the Filter group, you must configure both the Channel and Filter control tables.	**rmon channel index** <*index-number*> **port** <*port*> [**accept-type matched	failed**] [**data-control on	off**] [**turn-on-event-index** <*number*>] [**turn-off-event-index** <*number*>] [**event-index** <*number*>] [**channel-status ready	always-ready**][**description** <*string*>] [**owner** <*string*>] [**status enable	disable**] **rmon filter index** <*index-number*> **channel-index** <*number*> [**data-offset** <*number*>] [**data** <*string*>] [**data-mask** <*string*>] [**data-not-mask** <*string*>] [**pkt-status** <*number*>] [**status-mask** <*number*>] [**status-not-mask** <*number*>] [**owner** <*string*>] [**status enable	disable**]
To configure the Etherstats group.	**rmon etherstats index** <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable	disable**]
To configure the Event group.	**rmon event index** <*index-number*> **type none	log	trap	both** [**community** <*string*>] [**description** <*string*>] [**owner** <*string*>] [**status enable	disable**]
To configure the History group.	**rmon history index** <*index-number*> **port** <*port*> [**interval** <*seconds*>] [**owner** <*string*>] [**samples** <*num*>] [**status enable	disable**]

To configure the Application Layer and Network Layer Host groups.	**rmon hl-host index** <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable**	**disable**]
To configure the Application Layer and Network Layer Matrix groups.	**rmon hl-matrix index** <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable**	**disable**]
To configure the Host group.	**rmon host index** <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable**	**disable**]
To configure the Host Top N entries.	**rmon host-top-n index** <*index-number*> **host-index** <*number*> [**base** <*statistics*>] [**duration** <*time*>] [**size** <*size*>] [**owner** <*string*>] [**status enable**	**disable**]
To configure the Matrix group.	**rmon matrix index** <*index-number*> [**port** <*port*>] [**owner** <*string*>] [**status enable**	**disable**]
To configure the Network Layer Matrix top n entries.	**rmon nl-matrix-top-n index** <*index-number*> **matrix-index** <*number*> **ratebase terminal-packets**	**terminal-octets**	**all-packets**	**all-octets duration** <*number*> **size** <*number*> [**owner** <*string*>] [**status enable**	**disable**]
To configure the Protocol Distribution group.	**rmon protocol-distribution index** <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable**	**disable**]
To configure the User History group, you must configure the group of objects to be monitored and apply the objects in the group to the User History control table.	**rmon user-history-control index** <*index-number*> **objects** <*number*> **samples** <*number*> **interval** <*number*> [**owner** <*string*>] [**status enable**	**disable**] **rmon user-history-objects** <*groupname*> **variable** <*oid*> **type absolute**	**delta** [**status enable**	**disable**] **rmon user-history-apply** <*groupname*> **to** <*user-history-index*>

Configuration Examples

This section shows examples of configuration commands that specify an event that generates an SNMP trap and the alarm condition that triggers the event.

The RMON Alarm group allows the GSR to poll itself at user-defined intervals. Alarms that constitute an event are logged into the Event table that can then be polled by the management station. The management station is able to poll more network devices this way, as it only needs to poll the RMON Event table and not the device itself. The management station can also be sent trap information.

The following examples configure the GSR to create an event when a module is hot swapped into the chassis or any new IP interface is configured. The managed object ifTableLastChanged from RFC 2233) has an object identifier (OID) of 1.3.6.1.2.1.31.1.5.0 and the GSR will poll this OID every 5 minutes (300 seconds).

The command line below is an example of an RMON Event group configuration with the following attributes:

- Index number 15 to identify this entry in the Event control table.
- The event is both logged in the Event table and an SNMP trap generated with the community string "public".
- Event owner is "help desk".

```
gsr#(config) rmon event index 15 type both community public description
    "Interface added or module hot swapped in" owner "help desk"
```

The command line below is an example of an RMON Alarm group configuration with the following attributes:

- Index number 20 to identify this entry in the Alarm control table.
- The OID 1.3.6.1.2.1.31.1.5.0 identifies the attribute to be monitored.
- Samples taken at 300 second (5 minute) intervals.
- A "Startup" alarm generation condition instructing the GSR to generate an alarm if the sample is greater than or equal to the rising threshold or less than or equal to the falling threshold.
- Compare value at time of sampling (absolute value) to the specified thresholds.
- Rising and falling threshold values are 1.
- Rising and falling event index values are 15, which will trigger the previously-configured Event.

```
gsr#(config) rmon alarm index 20 variable 1.3.6.1.2.1.31.1.5.0 interval
    300 startup both type absolute-value rising-threshold 1 falling-
    threshold 1 rising-event-index 15 falling-event-index 15 owner "help
    desk"
```

Displaying RMON Information

The CLI **rmon show** commands allow you to display the same RMON statistics that can be viewed from a management station. To display RMON statistics for the GSR, use the following CLI command lines in Enable mode:

[1]To show Ethernet statistics.	**rmon show etherstats** <*port-list*> | **all-ports**
To show all events and logs.	**rmon show events**
To show all alarms.	**rmon show alarms**
To show histories and logs.	**rmon show history** <*port-list*> | **all-ports**
To show hosts and logs.	**rmon show hosts** <*port-list*> | **all-ports** [summary]
To show all Host Top N and logs.	**rmon show host-top-n**
To show matrices and logs.	**rmon show matrix** <*port-list*> | **all-ports**
To show all channels.	**rmon show channels**
To show all filters.	**rmon show filters**
To show all packet captures and logs.	**rmon show packet-capture**
To display the RMON 2 Protocol Directory.	**rmon show protocol-directory**
To display the RMON 2 Protocol Distribution.	**rmon show protocol-distribution** <*port-list*> | **all-ports**
To display the RMON 2 Address Map table.	**rmon show address-map** <*port-list*> | **all-ports**
To show Network Layer Host logs.	**rmon show nl-host**<*port-list*> | **all-ports [summary]**
To show Application Layer Host logs.	**rmon show al-host**<*port-list*> | **all-ports [summary]**
To show Network Layer Matrix logs.	**rmon show nl-matrix**<*port-list*> | **all-ports [order-by srcdst | dstsrc] [summary]**
To show Application Layer Matrix logs.	**rmon show al-matrix**<*port-list*> | **all-ports [order-by srcdst | dstsrc] [summary]**
To show all Network Layer Matrix Top N.	**rmon show nl-matrix-top-n**
To show all Application Layer Matrix Top N.	**rmon show al-matrix-top-n**

To show all user history logs.	**rmon show user-history**
To show probe configuration.	**rmon show probe-config [basic] [net-config] [trap-dest]**

[1]To display Ethernet statistics and related statistics for WAN ports, RMON has to be activated on that port. To activate RMON on a port, use the **frame-relay define service** or **ppp define service** command, and the **frame-relay apply service** or **ppp apply service** command. For additional information, refer to "Setting up a Frame Relay Service Profile" on page 304 (for frame relay) and "Setting up a PPP Service Profile" on page 308 (for PPP).

RMON CLI Filters

Because a large number of statistics can be collected for certain RMON groups, you can define and use CLI filters to limit the amount of information displayed with the **rmon show** commands. An RMON CLI filter can only be applied to a current Telnet or Console session.

The following shows Host table output *without* a CLI filter:

```
gs/r# rmon show hosts et.5.4
RMON I Host Table

Index: 503, Port: et.5.4, Owner: monitor
Address           InPkts   InOctets    OutPkts  OutOctets   Bcst   Mcst
-------           ------   --------    -------  ---------   ----   ----
00001D:921086          0          0        102       7140      0      0
00001D:9D8138       1128      75196        885     114387      0      0
00001D:A9815F          0          0        102       7140      0      0
00105A:08B98D          0          0        971     199960      0      0
004005:40A0CD          0          0         51       3264      0      0
006083:D65800          0          0       2190     678372      0      0
0080C8:E0F8F3          0          0        396      89818      0      0
00E063:FDD700          0          0        104      19382      0      0
01000C:CCCCCC       2188     678210          0          0      0      0
01005E:000009        204      14280          0          0      0      0
0180C2:000000       1519      97216          0          0      0      0
030000:000001        168      30927          0          0      0      0
080020:835CAA        885     114387       1128      75196      0      0
980717:280200          0          0       1519      97216      0      0
AB0000:020000          2        162          0          0      0      0
FFFFFF:FFFFFF       1354     281497          0          0      0      0
```

The following shows the same **rmon show hosts** command with a filter applied so that only hosts with inpkts greater than 500 are displayed:

```
gs/r#  rmon apply cli-filter 4
gs/r# rmon show hosts et.5.4
RMON I Host Table
Filter: inpkts > 500
Address       Port        InPkts   InOctets    OutPkts  OutOctets   Bcst   Mcst
-------       ----        ------   --------    -------  ---------   ----   ----
00001D:9D8138 et.5.4        1204      80110        941     121129      0      0
01000C:CCCCCC et.5.4        2389     740514          0          0      0      0
0180C2:000000 et.5.4        1540      98560          0          0      0      0
080020:835CAA et.5.4         940     121061       1204      80110      0      0
FFFFFF:FFFFFF et.5.4        1372     285105          0          0
```

RMON CLI filters can only be used with the following groups:

- Hosts
- Matrix
- Protocol Distribution
- Application Layer Host
- Network Layer Host
- Application Layer Matrix
- Network Layer Matrix

Creating RMON CLI Filters

To create RMON CLI filters, use the following CLI command in Configure mode:

Creates an RMON CLI filter.	**rmon set cli-filter** *<filter-id> <parameter>*

Using RMON CLI Filters

To see and use RMON CLI filters, use the following CLI command in User or Enable mode:

Displays RMON CLI filters.	**rmon show cli-filters**
Applies a CLI filter on current Telnet or Console session.	**rmon apply cli-filters** *<filter-id>*
Clears the currently-selected CLI filter.	**rmon clear cli-filter**

Troubleshooting RMON

If you are not seeing the information you expected with an **rmon show** command, or if the network management station is not collecting the desired statistics, first check that the port is up. Then, use the **rmon show status** command to check the RMON configuration on the GSR.

Check the following fields on the **rmon show status** command output:

```
gs/r# rmon show status
RMON Status
-----------
* RMON is ENABLED ❶
* RMON initialization successful.

             ❷
+--------------------------+
| RMON Group Status        |
+-------+--------+---------+
| Group | Status | Default |
+-------+--------+---------+
| Lite  |     On |     Yes |   ❹
+-------+--------+---------+
| Std   |     On |     Yes |
+-------+--------+---------+
| Pro   |     On |     Yes |
+-------+--------+---------+

RMON is enabled on: et.5.1, et.5.2, et.5.3, et.5.4, et.5.5, et.5.6, et.5.7, et.5.8 ❸

RMON Memory Utilization
-----------------------
        Total Bytes Available:   48530436

Total Bytes Allocated to RMON:    4000000
             Total Bytes Used:    2637872    ❺
             Total Bytes Free:    1362128
```

1. Make sure that RMON has been enabled on the GSR. When the GSR is booted, RMON is off by default. RMON is enabled with the **rmon enable** command.
2. Make sure that at least one of the RMON support levels—Lite, Standard, or Professional—is turned on with the **rmon set lite | standard | professional** command.
3. Make sure that RMON is enabled on the port for which you want statistics. Use the **rmon set ports** command to specify the port on which RMON will be enabled.
4. Make sure that the control table is configured for the report that you want. Depending upon the RMON group, default control tables may be created for all ports on the GSR. Or, if the RMON group is not one for which default control tables can be created, you will need to configure control table entries using the appropriate **rmon** command.

If you or your application are unable to crate a control table row, check the **snmp show status** output for errors. Make sure that there is a read-write community string. Verify that you can ping the GSR and that no ACLs prevent you from using SNMP to access the GSR.

5. Make sure that RMON has not run out of memory.

Allocating Memory to RMON

RMON allocates memory depending on the number of ports enabled for RMON, the RMON groups that have been configured, and whether or not default tables have been turned on or off. Enabling RMON with all groups (Lite, Standard, and Professional) with default tables uses approximately 300 Kbytes per port. If necessary, you can dynamically allocate additional memory to RMON.

To display the amount of memory that is currently allocated to RMON, use the following CLI command in Enable mode:

Displays the memory allocated to RMON.	**rmon show status**

Any memory allocation failures are reported. The following is an example of the information shown with the **rmon show status** command:

```
gs/r# rmon show status
RMON Status
-----------
* RMON is ENABLED
* RMON initialization successful.

+--------------------------+
| RMON Group Status |
+-------+--------+---------+
| Group | Status | Default |
+-------+--------+---------+
| Lite  |     On |     Yes |
+-------+--------+---------+
| Std   |     On |     Yes |
+-------+--------+---------+
| Pro   |     On |     Yes |
+-------+--------+---------+

RMON is enabled on: et.5.1, et.5.2, et.5.3, et.5.4, et.5.5, et.5.6,
et.5.7, et.5.8

RMON Memory Utilization
-----------------------
        Total Bytes Available:     48530436
Total Bytes Allocated to RMON:     4000000
             Total Bytes Used:     2637872
             Total Bytes Free:     1362128
```

To set the amount of memory allocated to RMON, use the following CLI command in User or Enable mode:

Specifies the total amount of Mbytes of memory allocated to RMON.	**rmon set memory** <*number*>

Chapter 22

WAN Configuration Guide

This chapter provides an overview of Wide Area Network (WAN) applications as well as an overview of both Frame Relay and PPP configuration for the GSR. In addition, you can view an example of a multi-router WAN configuration complete with diagram and configuration files in "WAN Configuration Examples" on page 313.

WAN Overview

On the DIGITAL GIGAswitch/Router, Wide Area Network (WAN) routing is performed over a serial interface using two basic protocols: Frame Relay and point-to-point protocol (PPP). Both protocols have their own set of configuration and monitoring CLI commands described in the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.

High-Speed Serial Interface (HSSI) and Standard Serial Interfaces

In both the Frame Relay and PPP environments on the GSR, you can specify ports to be High-Speed Serial Interface (HSSI) or standard serial interface ports, depending, of course, on the type of hardware you have. Each type of interface plays a part in the nomenclature of port identification. You must use either the "hs." or "se." prefix for HSSI and serial interfaces, respectively, when specifying WAN port identities.

For example, you would specify a frame relay serial WAN port located at router slot 4, port 1, on VC 100 as “se.4.1.100”.

Using the same approach, a PPP high-speed serial interface (HSSI) WAN port located at router slot 3, port 2 would be identified as “hs.3.2”.

Configuring WAN Interfaces

Configuring IP & IPX interfaces for the WAN is generally the same as for the LAN. You can configure IP/IPX interfaces on the physical port or you can configure the interface as part of a VLAN for WAN interfaces. However, in the case of IP interfaces, you can configure multiple IP addresses for each interface. Please refer to “Configuring IP Interfaces and Parameters” on page 61 and “Configuring IPX Interfaces and Parameters” on page 228 for more specific information.

There are some special considerations that apply only to WAN interfaces; these are detailed in this section.

Primary and Secondary Addresses

Like LAN interfaces, WAN interfaces can have primary and secondary IP addresses. For Frame Relay, you can configure primary and secondary addresses which are static or dynamic. For PPP, however, the primary addresses may be dynamic or static, but the secondary addresses must be static. This is because the primary addresses of both the local and peer routers are exchanged during IPCP/IPXCP negotiation.

Note: There is no mechanism in PPP for obtaining any secondary addresses from the peer.

Static, Mapped, and Dynamic Peer IP/IPX Addresses

The following sections describe the difference between static, mapped, and dynamic peer IP and IPX addresses and provide simple command line examples for configuration.

Static Addresses

If the peer IP/IPX address is known before system setup, you can specify the peer address when the interface is created. This disables Inverse ARP (InArp) for Frame Relay on that source/peer address pair; however, InArp will still be enabled for any other addresses on that interface or other interfaces. A static peer address for PPP means that the address the peer supplies during IP Control Protocol (IPCP) or IPX Control Protocol (IPXCP) negotiations will be ignored.

The following command line displays an example for a port:

```
interface create ip IPWAN address-netmask 10.50.1.1/16 peer-address
    10.50.1.2 port hs.3.1
```

The following command line displays an example for a VLAN:

```
interface create ip IPWAN address-netmask 10.50.1.1/16 peer-address
    10.50.1.2 vlan BLUE
```

Mapped Addresses

Mapped peer IP/IPX addresses are very similar to static addresses in that InArp is disabled for Frame Relay and the address negotiated in IPCP/IPXCP is ignored for PPP.

Mapped addresses are most useful when you do not want to specify the peer address using the **interface create** command. This would be the case if the interface is created for a VLAN and there are many peer addresses on the VLAN. If any of the peers on the VLAN do not support InArp or IPCP/IPXCP, then use a mapped address to configure the peer address.

The following command lines display two examples for Frame Relay:

```
frame-relay set peer-address ip-address 10.50.1.1/16 ports se.4.1.204
```

```
frame-relay set peer-address ipx-address a1b2c3d4.aa:bb:cc:dd:ee:ff
    ports se.6.3.16
```

The following command line displays two examples for PPP:

```
ppp set peer-address ip-address 10.50.1.1/16 ports se.4.1
```

```
ppp set peer-address ipx-address a1b2c3d4.aa:bb:cc:dd:ee:ff ports
    se.6.3
```

Dynamic Addresses

If the peer IP/IPX address is unknown, you do not need to specify it when creating the interface. When in the Frame Relay environment, the peer address will be automatically discovered via InArp. Similarly, the peer address will be automatically discovered via IPCP/IPXCP negotiation in a PPP environment.

The following command lines display examples for a port and a VC:

```
interface create ip IPWAN address-netmask 10.50.1.1/16 port hs.3.1
```

```
interface create ip IPWAN address-netmask 10.50.1.1/16 port hs.5.2.19
```

The following command line displays an example for a VLAN:

```
interface create ip IPWAN address-netmask 10.50.1.1/16 vlan BLUE
```

Forcing Bridged Encapsulation

WAN for the GSR has the ability to force bridged packet encapsulation. This feature has been provided to facilitate seamless compatibility with Cisco routers, which expect bridged encapsulation in certain operating modes.

The following command line displays an example for Frame Relay:

```
frame-relay set fr-encaps-bgd ports hs.5.2.19
```

The following command line displays an example for PPP:

```
ppp set ppp-encaps-bgd ports hs.5.2
```

Packet Compression

Packet compression can increase throughput and shorten the times needed for data transmission. You can enable packet compression for Frame Relay VCs and for PPP ports, however, both ends of a link must be configured to use packet compression.

Enabling compression on WAN serial links should be decided on a case by case basis. Important factors to consider include:

- average packet size
- nature of the data
- link integrity
- latency requirements

Each of these factors is discussed in more detail in the following sections and should be taken into consideration before enabling compression. Since the factors are dependent on the environment, you should first try running with compression histories enabled. If compression statistics do not show a very good long-term compression ratio, then select

the “no history” option. If the compression statistics do not improve or show a ration of less than 1, then compression should be disabled altogether.

Average Packet Size

In most cases, the larger the packet size, the better the potential compression ratio. This is due to the overhead involved with compression, as well as the compression algorithm. For example a link which always deals with minimum size packets may not perform as well as a link whose average packet size is much larger.

Nature of the Data

In general, data that is already compressed cannot be compressed any further. In fact, packets that are already compressed will grow even larger. For example, if you have a link devoted to streaming MPEG videos, you should *not* enable compression as the MPEG video data is already compressed.

Link Integrity

Links with high packet loss or links that are extremely over-subscribed may not perform as well with compression enabled. If this is the situation on your network, you should *not* enable compression histories (this applies only to PPP compressions; in Frame Relay compression, histories are always used).

Compression histories take advantage of data redundancy *between* packets. In an environment with high packet loss or over-subscribed links, there are many gaps in the packet stream resulting in very poor use of the compression mechanism. Compression histories work best with highly-correlated packet streams. Thus, a link with fewer flows will generally perform better than a link with many flows when compression histories are utilized.

The “no history” (max-histories = 0) option causes packets to be compressed on a packet-by-packet basis, thus packet loss is not a problem. Also, the number of flows is not an issue with this option as there is no history of previous packets.

Latency Requirements

The use of compression may affect a packet’s latency. Since the compressed packet is smaller, less time is needed to transmit it. On the other hand, each packet must undergo a compression/decompression process. Since the compression ratio will vary, the amount of latency will also vary.

Example Configurations

The following command line displays an example for Frame Relay:

```
frame-relay set payload-compress ports se.3.1.300
```

The following command line displays an example for PPP:

```
ppp set payload-compress port se.4.2
```

Packet Encryption

Packet encryption allows data to travel through unsecured networks. You can enable packet encryption for PPP ports, however, both ends of a link must be configured to use packet encryption.

The following command line displays an example:

```
ppp set payload-encrypt transmit-key 0x123456789abcdef receive-key
0xfedcba987654321 port se.4.2, mp.1
```

WAN Quality of Service

Increasing concentrations of audio, video, and data traffic are now presenting the networking industry with the significant challenge of employing the most effective use of WAN Quality-of-Service (QoS) as possible to ensure reliable end-to-end communication. For example, critical and time-sensitive traffic such as audio should have higher levels of bandwidth allocated than less time-sensitive traffic such as file transfers or e-mail. Simply adding more and more bandwidth to a network is not a viable solution to the problem. WAN access is extremely expensive, and there is a limited (albeit huge) supply. Therefore, making the most effective use of existing bandwidth is now a more critical issue than ever before.

The fact that IP communications to the desktop are clearly the most prevalent used today has made it the protocol of choice for end-to-end audio, video, and data applications. This means that the challenge for network administrators and developers has been to construct their networks to support these IP-based audio, video, and data applications along with their tried-and-true circuit-based applications over a WAN.

In addition, these audio, video, and data traffic transmissions hardly ever flow at a steady rate. Some periods will see relatively low levels of traffic, and others will temporarily surpass a firm's contracted Committed Information Rate (CIR). Carrier-based packet-switched networks such as Frame Relay and ATM are designed to handle these temporary peaks in traffic, but it is more cost- and resource- efficient to employ effective QoS configuration(s), thus relaxing the potential need to up your firm's CIR. By applying some

of the following sorts of attributes to interfaces on your network, you can begin to shape your network's QoS configuration to use existing bandwidth more effectively.

Source Filtering and ACLs

Source filtering and ACLs can be applied to a WAN interface; however, they affect the entire module, not an individual port.

For example, if you want to apply a source MAC address filter to a WAN serial card located in slot 5, port 2, your configuration command line would look like the following:

```
gs/r(config)# filters add address-filter name wan1 source-mac
    000102:030405 vlan 2 in-port-list se.5
```

Port se.5 is specified instead of se.5.2 because source filters affect the entire WAN module. Hence, in this example, **source-mac 000102:030405** would be filtered from ports se.5.1, se.5.2, se.5.3, and se.5.4 (assuming that you are using a four-port serial card).

ACLs work in a similar fashion. For example, if you define an ACL to deny all http traffic on one of the WAN interfaces, it will apply to the other WAN interfaces on that module as well. In practice, by making your ACLs more specific, for example by specifying source and destination IP addresses with appropriate subnet masks, you can achieve your intended level of control.

Weighted-Fair Queueing

Through the use of Weighted-Fair Queueing QoS policies, WAN packets with the highest priority can be allotted a sizable percentage of the available bandwidth and "whisked through" WAN interface(s). Meanwhile, the remaining bandwidth is distributed for "lower-priority" WAN packets according to the user's percentage-of-bandwidth specifications. Please refer to Chapter 35: "qos Commands" in the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual* for more detailed configuration information.

Note: Weighted-Fair Queueing applies only to best-effort traffic on the WAN card. If you apply any of the WAN specific traffic shaping commands, then weighted fair queuing will no longer be applicable.

Congestion Management

One of the most important features of configuring the GSR to ensure Quality of Service is the obvious advantage gained when you are able to avoid network congestion. The following topics touch on a few of the most prominent aspects of congestion avoidance when configuring the GSR.

Random Early Discard (RED)

RED allows network operators to manage traffic during periods of congestion based on policies. Random Early Discard (RED) works with TCP to provide fair reductions in traffic proportional to the bandwidth being used. Weighted Random Early Discard (WRED) works with IP Precedence or priority, as defined in the **qos** configuration command line, to provide preferential traffic handling for higher-priority traffic.

The CLI commands related to RED in both the Frame Relay and PPP protocol environments allow you to set maximum and minimum threshold values for each of the low-, medium-, and high-priority categories of WAN traffic.

Adaptive Shaping

Adaptive shaping implements the congestion-sensitive rate adjustment function and has the following characteristics:

- No blocking of data flow under normal condition if the traffic rate is below Bc+Be
- Reduction to a lower CIR upon detection of network congestion
- Progressive return to the negotiated information transfer rate upon congestion abatement

The CLI command related to adaptive shaping allows you to set threshold values for triggering the adaptive shaping function.

Frame Relay Overview

Frame relay interfaces are commonly used in a WAN to link several remote routers together via a single central switch. This eliminates the need to have direct connections between all of the remote members of a complex network, such as a host of corporate satellite offices. The advantage that Frame Relay offers to this type of geographic layout is the ability to switch packet data across the interfaces of different types of devices like switch-routers and bridges, for example.

Frame Relay employs the use of Virtual Circuits (VCs) when handling multiple logical data connections over a single physical link between different pieces of network equipment. The Frame Relay environment, by nature, deals with these connections quite well through its extremely efficient use of precious (sometimes scarce) bandwidth.

You can set up frame relay ports on your GSR with the commands described in Chapter 15: “frame-relay Commands” in the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.

Virtual Circuits

You can think of a Virtual Circuit (VC) as a "virtual interface" (sometimes referred to as "sub-interfaces") over which Frame Relay traffic travels. Frame Relay interfaces on the GSR use one or more VCs to establish bidirectional, end-to-end connections with remote end points throughout the WAN. For example, you can connect a series of multi-protocol routers in various locations using a Frame Relay network.

Permanent Virtual Circuits (PVCs)

WAN interfaces can take advantage of connections that assure a minimum level of available bandwidth at all times. These standing connections, called Permanent Virtual Circuits (PVCs), allow you to route critical packet transmissions from host to peer without concern for network congestion significantly slowing, let alone interrupting, your communications. PVCs are the most prevalent type of circuit used today and are similar to dedicated private lines in that you can lease and set them up through a service provider.

In a corporate setting, network administrators can use PVCs in an internal network to set aside bandwidth for critical connections, such as videoconferencing with other corporate departments.

Configuring Frame Relay Interfaces for the GSR

This section provides an overview of configuring a host of WAN parameters and setting up WAN interfaces. When working in the Frame Relay protocol environment, you must first define the type and location of the WAN interface. Having established the type and location of your WAN interfaces, you need to (optionally) define one or more service profiles for your WAN interfaces, then apply a service profile to the desired interface(s). An example of this process is covered in "Frame Relay Port Configuration" on page 305.

Defining the Type and Location of a Frame Relay and VC Interface

To configure a frame relay WAN port, you need to first define the type and location of one or more frame relay WAN ports or virtual circuits (VCs) on your GSR. The following command line displays a simplified example of a frame relay WAN port definition:

Define the type and location of a frame relay WAN port.	**port set** <*port*> **wan-encapsulation frame-relay speed** <*number*>

Note: If the port is a HSSI port that will be connected to a HSSI port on another router, you can also specify **clock** <*clock-source*> in your definition.

Then, you must set up a frame relay virtual circuit (VC). The following command line displays a simplified example of a VC definition:

Define the type and location of a frame relay VC.	**frame-relay create vc** *<port>*

Setting up a Frame Relay Service Profile

Once you have defined the type and location of your Frame Relay WAN interface(s), you can configure your GSR to more efficiently utilize available bandwidth for Frame Relay communications.

Note: The GSR comes with a set of "default values" for Frame Relay interface configuration settings, which means that setting up a Frame Relay service profile is not absolutely necessary to begin sending and receiving Frame Relay traffic on your GSR.

After you configure one or more service profiles for your Frame Relay interface(s), you can then apply a service profile to active Frame Relay WAN ports, specifying their behavior when handling Frame Relay traffic. The following command line displays all of the possible attributes used to define a Frame Relay service profile:

Define a frame relay service profile.	**frame-relay define service** *<service name>* **[Bc** *<number>***] [Be** *<number>***] [becn-adaptive-shaping** *<number>***] [cir** *<number>***] [high-priority-queue-depth** *<number>***] [low-priority-queue-depth** *<number>***] [med-priority-queue-depth** *<number>***] [red on|off] [red-maxTh-high-prio-traffic** *<number>***] [red-maxTh-low-prio-traffic** *<number>***] [red-maxTh-med-prio-traffic** *<number>***] [red-minTh-high-prio-traffic** *<number>***] [red-minTh-low-prio-traffic** *<number>***] [red-minTh-med-prio-traffic** *<number>***] [rmon on|off]**

Applying a Service Profile to an Active Frame Relay WAN Port

Once you have created one or more frame relay service profiles, you can specify their use on one or more active frame relay WAN ports on the GSR. The following command line displays a simplified example of this process:

Apply a service profile to an active WAN port.	**frame-relay apply service** *<service name>* **ports** *<port list>*

Monitoring Frame Relay WAN Ports

Once you have configured your frame relay WAN interface(s), you can use the CLI to monitor status and statistics for your WAN ports. The following table describes the monitoring commands for WAN interfaces, designed to be used in Enable mode:

Display a particular frame relay service profile	**frame-relay show service** <*service name*>
Display all available frame relay service profiles	**frame-relay show service all**
Display the last reported frame relay error	**frame-relay show stats port** <*port name*> **last-error**
Display active frame relay LMI parameters	**frame-relay show stats port** <*port name*> **lmi**
Display MIBII statistics for frame relay WAN ports	**frame-relay show stats port** <*port name*> **mibII**
Display a summary of all LMI statistics	**frame-relay show stats port** <*port name*> **summary**

Frame Relay Port Configuration

To configure frame relay WAN ports, you must first define the type and location of the WAN interface, optionally "set up" a library of configuration settings, then apply those settings to the desired interface(s). The following examples are designed to give you a small model of the steps necessary for a typical frame relay WAN interface specification.

To define the location and identity of a serial frame relay WAN port located at slot 5, port 1 with a speed rating of 45 million bits per second:

```
gs/r(config)# port set se.5.1 wan-encapsulation frame-relay speed
   45000000
```

To define the location and identity of a High-Speed Serial Interface (HSSI) VC located at slot 4, port 1 with a DLC of 100:

```
gs/r(config)# frame-relay create vc hs.4.1.100
```

Suppose you wish to set up a service profile called “profile1” that includes the following characteristics:

- Committed burst value of 2 million and excessive burst value of 1 million
- BECN active shaping at 65 frames
- Committed information rate (CIR) of 20 million bits per second
- Leave high-, low-, and medium-priority queue depths set to factory defaults
- Random Early Discard (RED) disabled
- RMON enabled

The command line necessary to set up a service profile with the above attributes would be as follows:

```
gs/r(config)# frame-relay define service profile1 Bc 2000000 Be 10000000
   becn-adaptive-shaping 65 cir 20000000 red off rmon on
```

To assign the above service profile to the VC interface created earlier (slot 4, port 1):

```
gs/r(config)# frame-relay apply service profile1 ports hs.4.1.100
```

Point-to-Point Protocol (PPP) Overview

Because of its ability to quickly and easily accommodate IP and IPX protocol traffic, Point-to-Point Protocol (PPP) routing has become a very important aspect of WAN configuration. Using PPP, you can set up router-to-router, host-to-router, and host-to-host connections.

Establishing a connection in a PPP environment requires that the following events take place:

- The router initializing the PPP connection transmits Link Control Protocol (LCP) configuration and test frames to the remote peer to set up the data link.
- Once the connection has been established, the router which initiated the PPP connection transmits a series of Network Control Protocol (NCP) frames necessary to configure one or more network-layer protocols.
- Finally, when the network-layer protocols have been configured, both the host and remote peer can send packets to one another using any and all of the configured network-layer protocols.

The link will remain active until explicit LCP or NCP frames instruct the host and/or the peer router to close the link, or until some external event (i.e., user interruption or system time-out) takes place.

You can set up PPP ports on your GSR with the commands described in Chapter 32: "port Commands" in the *DIGITAL GIGAswitch/Router Command Line Interface Reference Manual*.

Use of LCP Magic Numbers

LCP magic numbers enable you to detect situations where PPP LCP packets are looped back from the remote system, resulting in an error message. The use of LCP magic numbers is enabled on the GSR by default; however, should you employ a service profile in which the use of LCP magic numbers has been disabled, undetected "loopback" behavior may become a problem.

Note: In the event that a PPP WAN interface remains unrecognized at startup due to loopback interference, you can use the **ppp restart** command in the CLI to remedy the situation.

Configuring PPP Interfaces

This section provides an overview of configuring a host of WAN parameters and setting up WAN interfaces. When working in the PPP environment, you must first define the type and location of your WAN interfaces. Having established the type and location of your WAN interfaces, you need to (optionally) define one or more service profiles for your WAN interfaces, then apply a service profile to the desired interface(s). Examples of this process are displayed in "PPP Port Configuration" on page 311.

Defining the Type and Location of a PPP Interface

To configure a PPP WAN port, you need to first define the type and location of one or more PPP WAN ports on your GSR. The following command line displays a simplified example of a PPP WAN port definition:

Define the type and location of a PPP WAN port.	**port set** *<port>* **wan-encapsulation ppp speed** *<number>*

If the port is an HSSI port that will be connected to a HSSI port on another router, you can specify **clock** *<clock-source>* in the definition. (This feature is supported on HSSI boards, part number GSR-HSSI-02-AA.)

Setting up a PPP Service Profile

Once you have defined the type and location of your PPP WAN interface(s), you can configure your GSR to more efficiently utilize available bandwidth for PPP communications.

Note: The GSR comes with a set of "default values" for PPP interface configuration settings, which means that setting up a PPP service profile is not absolutely necessary to begin sending and receiving PPP traffic on your GSR.

After you configure one or more service profiles for your PPP interface(s), you can then apply a service profile to active PPP WAN ports, specifying their behavior when handling

PPP traffic. The following command line displays all of the possible attributes used to define a PPP service profile:

Define a PPP service profile.	**ppp define service** <*service name*> **[bridging enable|disable ip enable|disable ipx enable|disable] [high-priority-queue-depth** <*number*>**] [lcp-echo on|off] [lcp-magic on|off] [low-priority-queue-depth** <*number*>**] [max-configure** <*number*>**] [max-failure** <*number*>**] [max-terminate** <*number*>**] [med-priority-queue-depth** <*number*>**] [red on|off] [red-maxTh-high-prio-traffic** <*number*>**] [red-maxTh-low-prio-traffic** <*number*>**] [red-maxTh-med-prio-traffic** <*number*>**] [red-minTh-high-prio-traffic** <*number*>**] [red-minTh-low-prio-traffic** <*number*>**] [red-minTh-med-prio-traffic** <*number*>**] [retry-interval** <*number*>**] [rmon on|off]**

Note: If it is necessary to specify a value for Bridging, IP, and/or IPX, you must specify all three of these values at the same time. You cannot specify just one or two of them in the command line without the other(s).

Applying a Service Profile to an Active PPP Port

Once you have created one or more PPP service profiles, you can specify their use on one or more active PPP ports on the GSR. The following command line displays a simplified example of this process:

Apply a service profile to an active WAN port.	**ppp apply service** <*service name*> **ports** <*port list*>

Configuring Multilink PPP Bundles

The Multilink PPP (MLP) standard defines a method for grouping multiple physical PPP links into a logical pipe, called an "MLP bundle". Large packets are fragmented and transmitted over each physical link in an MLP bundle. At the destination, MLP reassembles the packets and places them in their correct sequence.

The following table describes the commands for configuring MLP:

Add PPP port(s) to an MLP bundle.	**ppp add-to-mlp** <*mlp* > **port** <*port list*>
Create MLP bundle(s).	**ppp create-mlp** <*mlp* > **slot** <*mlp* >
Set MLP encapsulation format.	**ppp set mlp-encaps-format ports** <*mlp* > **[format short-format]**
Set the size of frames that fragmented for transmission on an MLP bundle.	**ppp set mlp-frag-size ports** <*port list*> **size** <*size* >
Set the depth of the queue used to hold MLP packets for preserving the packet order.	**ppp set mlp-orderq-depth ports** <*port list*> **qdepth** <*number-of-packets*>
Set the depth of the queue used to hold packet fragments for reassembly.	**ppp set mlp-fragq-depth ports** <*port list*> **qdepth** <*number-of-packets*>

Compression on MLP Bundles or Links

Compression can be applied on either a bundle or link basis if MLP is enabled on PPP links. If compression is enabled on a bundle, the packets will be compressed *before* processing by MLP. If compression is enabled on a link, the packets will be compressed *after* the MLP processing.

In general, choose bundle compression over link compression whenever possible. Compressing packets before they are "split" by MLP is much more efficient for both the compression algorithm and the WAN card. Link compression is supported to provide the widest range of compatibility with other vendors' equipment.

Monitoring PPP WAN Ports

Once you have configured your PPP WAN interface(s), you can use the CLI to monitor status and statistics for your WAN ports. The following table describes the monitoring commands for WAN interfaces, designed to be used in Enable mode:

Display a particular PPP service profile.	**ppp show service** <*service name*>
Display all available PPP service profiles.	**ppp show service all**
Display bridge NCP statistics for specified PPP WAN port.	**ppp show stats port** <*port name*> **bridge-ncp**
Display IP NCP statistics for specified PPP WAN port.	**ppp show stats port** <*port name*> **ip-ncp**
Display link-status statistics for a specified PPP WAN port.	**ppp show stats port** <*port name*> **link-status**
Displays information for PPP ports that are added to MLP bundles.	**ppp show mlp** <*mlp list*>**|all-ports**

PPP Port Configuration

To configure PPP WAN ports, you must first define the type and location of the WAN interface, optionally "set up" a library of configuration settings, then apply those settings to the desired interface(s). The following examples are designed to give you a small model of the steps necessary for a typical PPP WAN interface specification.

To define the location and identity of a High-Speed Serial Interface (HSSI) PPP WAN port located at router slot 5, port 1 with a speed rating of 45 million bits per second:

```
gs/r(config)# port set hs.5.1 wan-encapsulation ppp speed 45000000
```

Suppose you wish to set up a service profile called "profile2" that includes the following characteristics:

- Bridging enabled
- Leave high-, low-, and medium-priority queue depths set to factory defaults
- IP and IPX enabled
- Sending of LCP Echo Requests disabled
- Use of LCP magic numbers disabled
- The maximum allowable number of unanswered requests set to 8
- The maximum allowable number of negative-acknowledgment transmissions set to 5
- The maximum allowable number of unanswered/improperly answered connection-termination requests before declaring the link to a peer lost set to 4
- Random Early Discard disabled
- The number of seconds between subsequent configuration request transmissions (the "retry interval") set to 25
- RMON enabled

The command line necessary to set up a service profile with the above attributes would be as follows:

```
gs/r(config)# ppp define service profile2 bridging enable ip enable ipx
   enable lcp-echo off lcp-magic off max-configure 8 max-failure 5 max-
   terminate 4 red off retry-interval 25 rmon on
```

To assign the above service profile to the active PPP WAN port defined earlier (slot 5, port 1):

```
gs/r(config)# ppp apply service profile2 ports hs.5.1
```

WAN Configuration Examples

Simple Configuration File

The following is an example of a simple configuration file used to test frame relay and PPP WAN ports:

```
port set hs.5.1 wan-encapsulation frame-relay speed 45000000
port set hs.5.2 wan-encapsulation ppp speed 45000000
interface create ip fr1 address-netmask 10.1.1.1/16 port hs.5.1.100
interface create ip ppp2 address-netmask 10.2.1.1/16 port hs.5.2
interface create ip lan1 address-netmask 10.20.1.1/16 port et.1.1
interface create ip lan2 address-netmask 10.30.1.1/16 port et.1.2
ip add route 10.10.0.0/16 gateway 10.1.1.2
ip add route 10.40.0.0/16 gateway 10.2.1.2
```

For a broader, more application-oriented WAN configuration example, see "Multi-Router WAN Configuration next.

Multi-Router WAN Configuration

The following is a diagram of a multi-router WAN configuration encompassing three subnets. From the diagram, you can see that R1 is part of both Subnets 1 and 2; R2 is part of both Subnets 2 and 3; and R3 is part of subnets 1 and 3. You can click on the router label (in blue) to jump to the actual text configuration file for that router:

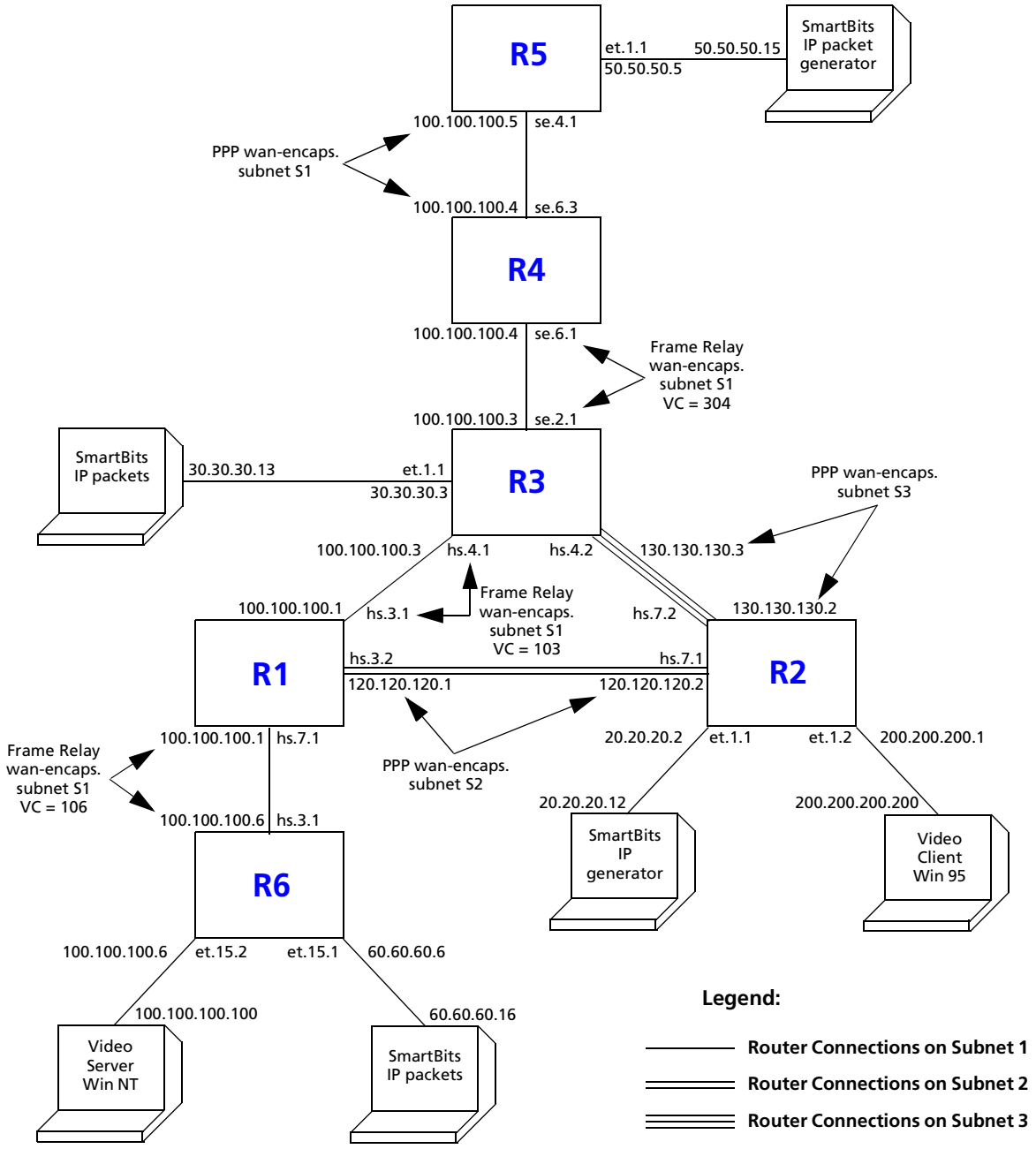


Figure 24. Multi-router WAN configuration

Router R1 Configuration File

The following configuration file applies to Router R1.

```
------------------------------------------------------------------------
Configuration for ROUTER R1
------------------------------------------------------------------------
port set hs.7.1 wan-encapsulation frame-relay speed 45000000
port set hs.3.1 wan-encapsulation frame-relay speed 45000000
port set hs.3.2 wan-encapsulation ppp speed 45000000
port set et.1.* duplex full

frame-relay create vc port hs.7.1.106
frame-relay create vc port hs.3.1.103
frame-relay define service CIRforR1toR6 cir 45000000 bc 450000
frame-relay apply service CIRforR1toR6 ports hs.7.1.106

vlan create s1 id 200
vlan create s2 id 300
vlan add ports hs.3.1.103,hs.7.1.106 to s1
vlan add ports hs.3.2 to s2
interface create ip s1 address-netmask 100.100.100.1/16 vlan s1
interface create ip s2 address-netmask 120.120.120.1/16 vlan s2

rip add interface all
rip set interface all version 2
rip set interface all xmt-actual enable
rip set auto-summary enable
rip start

system set name R1
```

Router R2 Configuration File

The following configuration file applies to Router R2.

```
------------------------------------------------------------------------
Configuration for ROUTER R2
------------------------------------------------------------------------
port set hs.7.1 wan-encapsulation ppp speed 45000000
port set hs.7.2 wan-encapsulation ppp speed 45000000
port set et.1.* duplex full

vlan create s2 id 300
interface create ip PPPforR2toR3 address-netmask 130.130.130.2/16 peer-address
    130.130.130.3 port hs.7.2
interface create ip SBitsLAN address-netmask 20.20.20.2/16 port et.1.1
vlan add ports hs.7.1 to s2
interface create ip s2 address-netmask 120.120.120.2/16 vlan s2
interface create ip VideoClient address-netmask 200.200.200.1/16 port et.1.2

qos set ip VideoFromNT high 100.100.100.100 200.200.200.200 any any
qos set ip VideoFrom95 high 200.200.200.200 100.100.100.100 any any

rip add interface all
rip set interface all version 2
rip set auto-summary enable
rip start

system set name R2
arp add 20.20.20.12 exit-port et.1.1 mac-addr 000202:020200
```

Router R3 Configuration File

The following configuration file applies to Router R3.

```
------------------------------------------------------------------------
Configuration for ROUTER R3
------------------------------------------------------------------------
port set se.2.1 wan-encapsulation frame-relay speed 1500000
port set et.1.* duplex full
port set hs.4.1 wan-encapsulation frame-relay speed 45000000
port set hs.4.2 wan-encapsulation ppp speed 45000000

frame-relay create vc port se.2.1.304
frame-relay create vc port hs.4.1.103

vlan create s1 id 200
interface create ip SBitsLAN address-netmask 30.30.30.3/16 port et.1.1
vlan add ports hs.4.1.103,se.2.1.304 to s1
interface create ip PPPforR2toR3 address-netmask 130.130.130.3/16 peer-address
    130.130.130.2 port hs.4.2
interface create ip s1 address-netmask 100.100.100.3/16 vlan s1

rip add interface all
rip set interface all version 2
rip set interface all xmt-actual enable
rip set broadcast-state always
rip set auto-summary enable
rip start

system set name R3

arp add 30.30.30.13 exit-port et.1.1 mac-addr 000303:030300
```

Router R4 Configuration File

The following configuration file applies to Router R4.

```
------------------------------------------------------------------------
Configuration for ROUTER R4
------------------------------------------------------------------------
port set se.6.1 wan-encapsulation frame-relay speed 1500000
port set se.6.3 wan-encapsulation ppp speed 1500000
port set et.1.* duplex full

frame-relay create vc port se.6.1.304

vlan create s1 id 200
vlan add ports se.6.1.304,se.6.3 to s1
interface create ip s1 address-netmask 100.100.100.4/16 vlan s1

rip add interface all
rip set interface all version 2
rip set interface all xmt-actual enable
rip set broadcast-state always
rip set auto-summary enable
rip start

system set name R4
```

Router R5 Configuration File

The following configuration file applies to Router R5.

```
------------------------------------------------------------------------
Configuration for ROUTER R5
------------------------------------------------------------------------
port set se.4.1 wan-encapsulation ppp speed 1500000
port set et.1.* duplex full

vlan create s1 id 200

interface create ip lan1 address-netmask 50.50.50.5/16 port et.1.1
vlan add ports se.4.1 to s1
interface create ip s1 address-netmask 100.100.100.5/16 vlan s1

rip add interface all
rip set auto-summary enable
rip set interface all version 2
rip start

system set name R5

arp add 50.50.50.15 mac-addr 000505:050500 exit-port et.1.1
```

Router R6 Configuration File

The following configuration file applies to Router R6.

```
------------------------------------------------------------------------
Configuration for ROUTER R6
------------------------------------------------------------------------
port set et.15.* duplex full
port set hs.3.1 wan-encapsulation frame-relay speed 45000000

frame-relay create vc port hs.3.1.106
frame-relay define service CIRforR1toR6 cir 45000000 bc 450000
frame-relay apply service CIRforR1toR6 ports hs.3.1.106

vlan create BridgeforR1toR6 port-based id 106
interface create ip FRforR1toR6 address-netmask 100.100.100.6/16 vlan
BridgeforR1toR6
interface create ip lan1 address-netmask 60.60.60.6/16 port et.15.1
vlan add ports hs.3.1.106 to BridgeforR1toR6
vlan add ports et.15.2 to BridgeforR1toR6

qos set ip VideoFromNT high 100.100.100.100 200.200.200.200 any any
qos set ip VideoFrom95 high 200.200.200.200 100.100.100.100 any any

rip add interface all
rip set interface all version 2
rip set auto-summary enable
rip start

system set name R6

arp add 60.60.60.16 mac-addr 000606:060600 exit-port et.15.1
```

9032684-03 Printed in U.S.A.