SmartSwitch Router
Command Line Interface
Reference Manual

9032553-05

CABLETRON SYSTEMS

Notice

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

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

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

Cabletron Systems, Inc.
35 Industrial Way
Rochester, NH 03867-5005

Printed in the United States of America

Order Number: 9032553-05

LANVIEW is a registered trademark, and **SmartSwitch** is a trademark of Cabletron Systems, Inc.

CompuServe is a registered trademark of CompuServe, Inc.

i960 microprocessor is a registered trademark of Intel Corp.

Ethernet is a trademark of Xerox Corporation.

FCC Notice

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment uses, generates, and can radiate radio frequency energy and if not installed in accordance with the operator's manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause interference in which case the user will be required to correct the interference at his own expense.

WARNING: Changes or modifications made to this device which are not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

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 documents (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

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may arise. When such trouble occurs, the user may be required to take corrective actions.

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。

CABLETRON SYSTEMS, INC.
PROGRAM LICENSE AGREEMENT

IMPORTANT: THIS LICENSE APPLIES FOR USE OF PRODUCT IN THE FOLLOWING GEOGRAPHICAL REGIONS:

CANADA
MEXICO
CENTRAL AMERICA
SOUTH AMERICA

BEFORE OPENING OR UTILIZING THE ENCLOSED PRODUCT, CAREFULLY READ THIS LICENSE AGREEMENT.

This document is an agreement ("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 ("Program") in the package. The Program may be contained in firmware, chips or other media. 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, RETURN THE UNOPENED PRODUCT TO CABLETRON OR YOUR DEALER, IF ANY, WITHIN TEN (10) DAYS FOLLOWING THE DATE OF RECEIPT FOR A FULL REFUND.

IF YOU HAVE ANY QUESTIONS ABOUT THIS AGREEMENT, CONTACT CABLETRON SYSTEMS (603) 332-9400. Attn: Legal Department.

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.

4. **EXPORT REQUIREMENTS.** You understand that Cabletron and its Affiliates are subject to regulation by agencies of the U.S. Government, including the U.S. Department of Commerce, which prohibit export or diversion of certain technical products to certain countries, unless a license to export the product is obtained from the U.S. Government or an exception from obtaining such license may be relied upon by the exporting party.

   If the Program is exported from the United States pursuant to the License Exception CIV under the U.S. Export Administration Regulations, You agree that You are a civil end user of the Program and agree that You will use the Program for civil end uses only and not for military purposes.

   If the Program is exported from the United States pursuant to the License Exception TSR under the U.S. Export Administration Regulations, in addition to the restriction on transfer set forth in

Sections 1 or 2 of this Agreement, You agree not to (i) reexport or release the Program, the source code for the Program or technology to a national of a country in Country Groups D:1 or E:2 (Albania, Armenia, Azerbaijan, Belarus, Bulgaria, Cambodia, Cuba, Estonia, Georgia, Iraq, Kazakhstan, Kyrgyzstan, Laos, Latvia, Libya, Lithuania, Moldova, North Korea, the People's Republic of China, Romania, Russia, Rwanda, Tajikistan, Turkmenistan, Ukraine, Uzbekistan, Vietnam, or such other countries as may be designated by the United States Government), (ii) export to Country Groups D:1 or E:2 (as defined herein) the direct product of the Program or the technology, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List, or (iii) if the direct product of the technology is a complete plant o r any major component of a plant, export to Country Groups D:1 or E:2 the direct product of the plant or a major component thereof, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List or is subject to State Department controls under the U.S. Munitions List.

5. **UNITED STATES GOVERNMENT RESTRICTED RIGHTS.** The enclosed Product (i) was developed solely at private expense; (ii) contains "restricted computer software" submitted with restricted rights in accordance with section 52.227-19 (a) through (d) of the Commercial Computer Software-Restricted Rights Clause and its successors, and (iii) in all respects is proprietary data belonging to Cabletron and/or its suppliers. For Department of Defense units, the Product is considered commercial computer software in accordance with DFARS section 227.7202-3 and its successors, and use, duplication, or disclosure by the Government is subject to restrictions set forth herein.

6. **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 EXPRESS 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.

7. **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 IN THE DURATION OR LIMITATION OF IMPLIED WARRANTIES IN SOME INSTANCES, THE ABOVE LIMITATION AND EXCLUSIONS MAY NOT APPLY TO YOU.

CABLETRON SYSTEMS SALES AND SERVICE, INC. PROGRAM LICENSE AGREEMENT

IMPORTANT: THIS LICENSE APPLIES FOR USE OF PRODUCT IN THE UNITED STATES OF AMERICA AND BY UNITED STATES OF AMERICA GOVERNMENT END USERS.

BEFORE OPENING OR UTILIZING THE ENCLOSED PRODUCT, CAREFULLY READ THIS LICENSE AGREEMENT.

This document is an agreement ("Agreement") between You, the end user, and Cabletron Systems Sales and Service, Inc. ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software program ("Program") in the package. The Program may be contained in firmware, chips or other media. 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, RETURN THE UNOPENED PRODUCT TO CABLETRON OR YOUR DEALER, IF ANY, WITHIN TEN (10) DAYS FOLLOWING THE DATE OF RECEIPT FOR A FULL REFUND.

IF YOU HAVE ANY QUESTIONS ABOUT THIS AGREEMENT, CONTACT CABLETRON SYSTEMS (603) 332-9400. Attn: Legal Department.

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.

4. **EXPORT REQUIREMENTS.** You understand that Cabletron and its Affiliates are subject to regulation by agencies of the U.S. Government, including the U.S. Department of Commerce, which prohibit export or diversion of certain technical products to certain countries, unless a license to export the product is obtained from the U.S. Government or an exception from obtaining such license may be relied upon by the exporting party.

   If the Program is exported from the United States pursuant to the License Exception CIV under the U.S. Export Administration Regulations, You agree that You are a civil end user of the Program and agree that You will use the Program for civil end uses only and not for military purposes.

   If the Program is exported from the United States pursuant to the License Exception TSR under the U.S. Export Administration Regulations, in addition to the restriction on transfer set forth in Sections 1 or 2 of this Agreement, You agree not to (i) reexport or release the Program, the source code for the Program or technology to a national of a country in Country Groups D:1 or E:2 (Albania, Armenia, Azerbaijan, Belarus, Bulgaria, Cambodia, Cuba, Estonia, Georgia, Iraq,

Kazakhstan, Kyrgyzstan, Laos, Latvia, Libya, Lithuania, Moldova, North Korea, the People's Republic of China, Romania, Russia, Rwanda, Tajikistan, Turkmenistan, Ukraine, Uzbekistan, Vietnam, or such other countries as may be designated by the United States Government), (ii) export to Country Groups D:1 or E:2 (as defined herein) the direct product of the Program or the technology, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List, or (iii) if the direct product of the technology is a complete plant o r any major component of a plant, export to Country Groups D:1 or E:2 the direct product of the plant or a major component thereof, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List or is subject to State Department controls under the U.S. Munitions List.

5. **UNITED STATES GOVERNMENT RESTRICTED RIGHTS.** The enclosed Product (i) was developed solely at private expense; (ii) contains "restricted computer software" submitted with restricted rights in accordance with section 52.227-19 (a) through (d) of the Commercial Computer Software-Restricted Rights Clause and its successors, and (iii) in all respects is proprietary data belonging to Cabletron and/or its suppliers. For Department of Defense units, the Product is considered commercial computer software in accordance with DFARS section 227.7202-3 and its successors, and use, duplication, or disclosure by the Government is subject to restrictions set forth herein.

6. **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 EXPRESS 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.

7. **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 IN THE DURATION OR LIMITATION OF IMPLIED WARRANTIES IN SOME INSTANCES, THE ABOVE LIMITATION AND EXCLUSIONS MAY NOT APPLY TO YOU.

CABLETRON SYSTEMS LIMITED PROGRAM LICENSE AGREEMENT

IMPORTANT: THIS LICENSE APPLIES FOR THE USE OF THE PRODUCT IN THE FOLLOWING GEOGRAPHICAL REGIONS:

EUROPE
MIDDLE EAST
AFRICA
ASIA
AUSTRALIA
PACIFIC RIM

BEFORE OPENING OR UTILIZING THE ENCLOSED PRODUCT, CAREFULLY READ THIS LICENSE AGREEMENT.

This document is an agreement ("Agreement") between You, the end user, and Cabletron Systems Limited ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software program ("Program") in the package. The Program may be contained in firmware, chips or other media. 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, RETURN THE UNOPENED PRODUCT TO CABLETRON OR YOUR DEALER, IF ANY, WITHIN TEN (10) DAYS FOLLOWING THE DATE OF RECEIPT FOR A FULL REFUND.

IF YOU HAVE ANY QUESTIONS ABOUT THIS AGREEMENT, CONTACT CABLETRON SYSTEMS (603) 332-9400. Attn: Legal Department.

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 governed in accordance with English law. The English courts shall have exclusive jurisdiction in the event of any disputes.

4. **EXPORT REQUIREMENTS.** You understand that Cabletron and its Affiliates are subject to regulation by agencies of the U.S. Government, including the U.S. Department of Commerce, which prohibit export or diversion of certain technical products to certain countries, unless a license to export the product is obtained from the U.S. Government or an exception from obtaining such license may be relied upon by the exporting party.

   If the Program is exported from the United States pursuant to the License Exception CIV under the U.S. Export Administration Regulations, You agree that You are a civil end user of the Program and agree that You will use the Program for civil end uses only and not for military purposes.

If the Program is exported from the United States pursuant to the License Exception TSR under the U.S. Export Administration Regulations, in addition to the restriction on transfer set forth in Sections 1 or 2 of this Agreement, You agree not to (i) reexport or release the Program, the source code for the Program or technology to a national of a country in Country Groups D:1 or E:2 (Albania, Armenia, Azerbaijan, Belarus, Bulgaria, Cambodia, Cuba, Estonia, Georgia, Iraq, Kazakhstan, Kyrgyzstan, Laos, Latvia, Libya, Lithuania, Moldova, North Korea, the People's Republic of China, Romania, Russia, Rwanda, Tajikistan, Turkmenistan, Ukraine, Uzbekistan, Vietnam, or such other countries as may be designated by the United States Government), (ii) export to Country Groups D:1 or E:2 (as defined herein) the direct product of the Program or the technology, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List, or (iii) if the direct product of the technology is a complete plant o r any major component of a plant, export to Country Groups D:1 or E:2 the direct product of the plant or a major component thereof, if such foreign produced direct product is subject to national security controls as identified on the U.S. Commerce Control List or is subject to State Department controls under the U.S. Munitions List.

5. **UNITED STATES GOVERNMENT RESTRICTED RIGHTS.** The enclosed Product (i) was developed solely at private expense; (ii) contains "restricted computer software" submitted with restricted rights in accordance with section 52.227-19 (a) through (d) of the Commercial Computer Software-Restricted Rights Clause and its successors, and (iii) in all respects is proprietary data belonging to Cabletron and/or its suppliers. For Department of Defense units, the Product is considered commercial computer software in accordance with DFARS section 227.7202-3 and its successors, and use, duplication, or disclosure by the Government is subject to restrictions set forth herein.

6. **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 EXPRESS 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.

7. 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 IN THE DURATION OR LIMITATION OF IMPLIED WARRANTIES IN SOME INSTANCES, THE ABOVE LIMITATION AND EXCLUSIONS MAY NOT APPLY TO YOU.

SAFETY INFORMATION

CLASS 1 LASER TRANSCEIVERS

The SSR-HFX11-08 100Base-FX Module, SSR-GSX11-02 1000Base-LX Module, SSR-GLX19-02 1000Base-LX Module, SSR-HFX29-08 100Base-FX SMF Module, SSR-GLX70-01 1000Base-LLX module, SSR-2-SX 1000Base-SX Module, SSR-2-LX 1000Base-LX Module, SSR-2-LX70 1000Base-LX Module, and SSR-2-GSX system 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.

SAFETY INFORMATION

CLASS 1 LASER TRANSCEIVERS

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 x 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 x 10^{3} W m^{2} sr-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 ADDENDUM

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

About This Manual

This manual provides reference information for the commands in the SmartSwitch Router (SSR) Command Line Interface (CLI). For product information not available in this manual, see the manuals listed in "Related Documentation" on page 31.

Note: If you plan to use Cabletron CoreWatch to configure or manage the SSR, see the *CoreWatch User's Manual* and the CoreWatch online help for information.

Who Should Read This Manual?

Read this manual if you are a network administrator responsible for configuring or managing the SSR.

How to Use This Manual

The CLI commands and facilities are organized alphabetically in this manual. To locate information about a command, go to the chapter for the command or for the facility that contains the command. For example, to find information about the **configure** command, go to "configure Command" on page 147. To find information about the **interface add** command, go to "interface Commands" on page 251, then locate the description of the **interface add** command within that chapter.

Related Documentation

The SSR 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 SSR	*SmartSwitch Router Getting Started Guide*
Managing the SSR using the CoreWatch Web-based management application	*CoreWatch User's Manual* and the CoreWatch online help

For Information About...	See the...
How to use CLI (Command Line Interface) commands to configure and manage the SSR	*SmartSwitch Router User Reference Manual*
SYSLOG messages and SNMP traps	*SmartSwitch Router Error Reference Manual*

CLI Parameter Types

The following table describes all the parameter types you can use with the CLI.

Data Type	Descripton	Example
conditional	A numerical conditional expression. Special symbols are used to describe a numerical condition: > (greater than), < (less than) and != (not equal to).	<1024 or >2048 or !=4096
hexadecimal	A hexadecimal number	a7 or 0xa7
hostname	Hostname of an IP host	gauguin or john-pc
hostname/IP	Hostname or IP address of a host	nagasaki or 10.43.1.4
keyword	A keyword described in the list of acceptable keywords in the online help	on or off
interface name or IP address	Name of an interface or its IP address	int1 or 10.1.4.33
interface name list	A list of one or more interface names delimited by commas	int1 or int1,int2,int3
IP address	An IP address of the form x.x.x.x. Some commands may explicitly require a unicast or multicast address.	10.1.2.3
IP address/mask	A pair of IP address and mask values. Depending on the command, the mask may be a network mask or filtering mask. The mask can be described using the traditional IP address syntax (255.0.0.0) or a CIDR syntax (/8).	10.1.4.0/255.255.255.0 or 10.1.4.0/24
IP address list	A list of IP addresses separated by spaces but enclosed in quotes.	“10.1.4.4 10.1.5.5 10.1.6.6”

Data Type	Descripton	Example
IPX network address	An IPX network address in hexadecimal	
IPX network.node address	An IPX network and node address of the form <netaddr>.<macaddr> where <netaddr> is the network address of a host and <macaddr> is the node or MAC address of the IPX host. For some commands, if the node address is not given, the node address is assumed to be a wildcard.	a1b2c3d4.0820a1:f3:38:11 or aa89f383
IPX SAP server name	An alphanumeric string representing a valid IPX SAP server name where the following characters are illegal: “*./:;<=>?[]\ |	server1
MAC address	A MAC address specified in one of two forms: xx:xx:xx:xx:xx:xx or xxxxxx:xxxxxx	08:00:50:1a:2b:c3 or 080050:1a2bc3
number	An integer number	100
numerical range	A number or a range of numbers	5 or 7-10
port	A single port	et.1.4, gi.2.1, hs.3.1.100, or se.4.2.200
port list	A list of one or more ports. To specify a range of ports within a module, describe the range in parenthesis. You can also specify non-consecutive ports by using commas to separate them. The wildcard character (*) can also be used to specify all modules or all ports within a module	et.1.(3-8) or et.1.(1,3,5), hs.(1-2).1.100, or se.4.(1-3).200, gi.2.*
slot number	A list of one or more occupied slots in the SSR	1 or 7

Data Type	Descripton	Example
string	A character string. To include spaces in a string, specify the entire string in double quotes (“).	abc or “abc def”
URL	A Uniform Resource Locator. The type of URL depends on the command where the URL is used. Currently, two URLs are supported: TFTP: *tftp://host/pathname* RCP: *rcp://username@host/pathname*	tftp://10.1.4.5/test/abc.txt rcp://dave@rtr/test/abc.txt

Chapter 1

acl Commands

The acl commands allow you to create ACLs (Access Control Lists) and apply them to IP and IPX interfaces on the SSR. An ACL permits or denies switching of packets based on criteria such as the packet's source address and destination address, TCP or UDP port number, and so on. When you apply an ACL to an interface, you can specify whether the ACL affects incoming traffic or outgoing traffic. You also can enable a log of the ACL's use.

Command Summary

Table 1 lists the acl commands. The sections following the table describe the command syntax.

Table 1. acl commands

acl <*name*> **apply interface** <*InterfaceName*> **input** | **output** **[logging on** | **off** | **deny-only** | **permit-only][policy local** | **external]**
acl <*name*> **apply service** <*ServiceName*> **[logging [on** | **off]]**
acl <*name*> **permit** | **deny icmp** <*SrcAddr/Mask*> <*DstAddr/Mask*>
acl <*name*> **permit** | **deny igmp** <*SrcAddr/Mask*> <*DstIP/mask*>
acl <*name*> **permit** | **deny ip** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> **[accounting]**
acl <*name*> **permit** | **deny ip-protocol** <*proto-num*> <*SrcAddr/Mask*> <*DstAddr/Mask*> <*tos*>
acl <*name*> **permit** | **deny ipx** <*SrcAddr*> <*SrcSocket*> <*DstAddr*> <*DstSocket*> <*SrcNetMask*> <*DstNetMask*>
acl <*name*> **permit** | **deny ipxgns** <*ServerAddr*> <*ServiceType*> <*ServiceName*>

Table 1. acl commands (Continued)

acl <*name*> **permit** | **deny ipxrip** <*FromNetwork*> <*ToNetwork*>
acl <*name*> **permit** | **deny ipxsap** <*ServerAddr*> <*ServiceType*> <*ServiceName*>
acl <*name*> **permit** | **deny ipxtype20**
acl <*name*> **permit** | **deny tcp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> [**accounting**][**established**]
acl <*name*> **permit** | **deny udp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> [**accounting**]
acl-policy enable external

acl apply interface

Purpose

Apply an ACL to an interface.

Format

acl <*name*> **apply interface** <*InterfaceName*> **input | output**
[logging on | off | deny-only | permit-only] [policy local | external]

Mode

Configure

Description

The **acl apply interface** command applies a previously defined ACL to an interface. When you apply an ACL to an interface, you implicitly enable access control on that interface. You can apply an ACL to filter out inbound traffic, outbound traffic, or both inbound and outbound traffic. Inbound traffic is packets coming into the interface while outbound traffic is packets going out of that interface.

When you apply an ACL, you also can enable ACL Logging by using the **logging** keyword. When you enable ACL Logging on an interface, the SSR displays ACL Logging messages on the console. The ACL log provides information such as the interface name, the ACL name, whether the packet is forwarded or not, and the internal details of the packet.

You can also specify if the ACL is allowed to be modified or removed from the interface by an external agent (such as a policy manager application) by using the **policy** keyword. If you do not specify the **policy** keyword, an external agent is allowed to modify or remove the applied ACL. Note that the **acl-policy enable external** command must be in the configuration before an external agent can modify or remove an applied ACL.

Parameters

<*name*>	Name of the ACL. The ACL must already be defined. To define an ACL, use one of the commands described in other sections in this chapter.
<*InterfaceName*>	Name of the interface to which you are applying the ACL.

input Applies the ACL to filter out inbound traffic.

output Applies the ACL to filter out outbound traffic.

logging on | off | deny-only | permit-only
Enables or disables ACL logging for this interface. You can specify one of the following keywords:

- **off** Disables all logging.
- **on** Enables logging of packets that are dropped or forwarded because of ACL.
- **deny-only** Enables logging of dropped packets only.
- **permit-only** Enables logging of forwarded packets only.

policy local | external
Allows or prevents an external agent from modifying or removing the applied ACL. You can specify one of the following keywords:

- **local** External agent cannot modify or remove the applied ACL.
- **external** External agent can modify or remove the applied ACL. This is the default.

Restrictions

You can apply only one ACL of each type (IP or IPX) to an interface at one time. For example, although you can define two ACLs, "ipacl1" and "ipacl2", you cannot apply them both to the same interface.

You can apply IP ACLs only to IP interfaces. Likewise, you can apply IPX ACLs only to IPX interfaces.

Examples

To apply ACL "100" to interface *int4* to filter out inbound traffic:

```
ssr(config)# acl 100 apply interface int4 input
```

To apply ACL "nonfs" to interface *int16* to filter out outbound traffic and enable logging:

```
ssr(config)# acl nonfs apply interface int16 output logging on
```

acl apply service

Purpose

Apply an ACL to a service on the SSR.

Format

acl <*name*> **apply service** <*ServiceName*> **[logging [on | off]]**

Mode

Configure

Description

The **acl apply service** command applies a previously defined ACL to a service provided by the SSR. A service is typically a server or agent running on the SSR, for example, a Telnet server or SNMP agent. By applying an ACL to a service, you can control which host can access individual services on the SSR. This type of ACL is known as a Service ACL. It does not control packets going *through* the SSR. It only controls packets that are *destined* for the SSR, specifically, one of the services provided by the SSR. As a result, a Service ACL, by definition, is applied only to check for inbound traffic to the SSR. In addition, if a Service ACL is defined with destination address and port information, that information is ignored. The destination host of a Service ACL is by definition the SSR. The destination port is the well-known port of the service.

When you apply an ACL, you also can enable ACL Logging by using the **logging** keyword. When you enable ACL Logging on an interface, the SSR displays ACL Logging messages on the console. The ACL log provides information such as the interface name, the ACL name, whether the packet is forwarded or not, and the internal details of the packet.

Parameters

<*name*>	Name of the Service ACL. The ACL must already be defined. To define an ACL, use one of the commands described in other sections in this chapter.
<*ServiceName*>	Name of the service on the SSR to which you are applying the ACL. Currently, the following services are supported:

http HTTP web server

snmp SNMP agent

telnet Telnet server

[logging [on | off]] Enables or disables ACL logging for this interface. You can specify one of the following keywords:

off Disables logging.

on Enables logging.

Restrictions

You can apply only one ACL of each type (IP or IPX) to a service at one time. For example, although you can define two ACLs, “ipacl1” and “ipacl2”, you cannot apply them both to the same service.

Examples

To permit access to the SNMP agent only from the host 10.4.3.33 (presumably an SNMP management station):

```
ssr(config)# acl 100 permit udp 10.4.3.33
ssr(config)# acl 100 apply service snmp
```

The following commands permit access to the Telnet server from hosts on the subnet 10.4.7.0/24 with a privileged source port. In addition, with logging enabled, all incoming Telnet accesses are logged to the console.

```
ssr(config)# acl 120 permit tcp 10.4.7.0/24 <1024
ssr(config)# acl 120 apply service telnet logging on
```

The following commands permit access to the HTTP web server from subnet 10.12.4.0/24. Notice that even though the destination address and port are specified for this ACL (*10.12.7.44* and *any* port), they are ignored. This service ACL will match only packets destined for the SSR itself and the well-known port of the service (port 80 for HTTP).

```
ssr(config)# acl 140 permit ip 10.12.4.0/24 any 10.12.7.44 any
ssr(config)# acl 120 apply service http
```

acl permit|deny icmp

Purpose

Create an ICMP ACL.

Format

acl <*name*> **permit | deny icmp** <*SrcAddr/Mask*> <*DstAddr/Mask*>

Mode

Configure

Description

The **acl permit icmp** and **acl deny icmp** commands define an ACL to allow or block ICMP traffic from entering or leaving the SSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the SSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*SrcAddr/Mask*>	The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").
<*DstAddr/Mask*>	The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To deny ICMP traffic from the subnet 10.24.5.0 (with a 24 bit netmask) to any destination:

```
ssr(config)# acl 310 deny icmp 10.24.5.0/24 any
```

To create an ACL to permit ICMP traffic from the host 10.12.28.44 to subnet 10.43.21.0:

```
ssr(config)# acl 312 permit icmp 10.12.28.44 10.43.21.0/24
```

acl permit|deny igmp

Purpose

Create an IGMP ACL.

Format

acl <*name*> **permit | deny igmp** <*SrcAddr/Mask*> <*DstAddr/Mask*>

Mode

Configure

Description

The **acl permit igmp** and **acl deny igmp** commands define an ACL to allow or block IGMP traffic from entering or leaving the SSR. For each of the values describing a flow, you can use the keyword **any** to specify a wildcard ("don't care") condition. If you do not specify a value for a field, the SSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword.

<*name*> Name of this ACL. You can use a string of characters or a number.

<*SrcAddr/Mask*> The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

<*DstAddr/Mask*> The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to deny IGMP traffic from the subnet 10.1.5.0 (with a 24 bit netmask) to any destination:

```
ssr(config)# acl 410 deny igmp 10.1.5.0/24 any
```

To create an ACL to permit IGMP traffic from the host 10.33.34.44 to subnet 10.11.21.0:

```
ssr(config)# acl 714 permit igmp 10.33.34.44 10.11.21.0/24
```

acl permit|deny ip

Purpose

Create an IP ACL.

Format

acl <*name***> permit | deny ip** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> [**accounting**]

Mode

Configure

Description

The **acl permit ip** and **acl deny ip** commands define an Access Control List to allow or block IP traffic from entering or leaving the router. Unlike the more specific variants of the acl commands for **tcp** and **udp**, the IP version of the command includes IP-based protocols such as **tcp**, **udp**, **icmp** and **igmp**. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the SSR assumes that the value is a wildcard (as if you had specified the **any** keyword). The two exceptions to this rule are the optional parameters <*tos*> (type of service) and **accounting**. <*tos*> is a value from 0 to 15. The **accounting** keyword is only valid for the **permit** command, and can be placed anywhere on the command line. When you specify the **accounting** keyword, LFAP accounting information will be sent to the configured server for flows that match the ACL.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*SrcAddr/Mask*>	The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

<*DstAddr/Mask*>	The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.
<*SrcPort*>	For TCP or UDP, the number of the source TCP or UDP port. This field applies only to TCP or UDP traffic. If the incoming packet is ICMP or another non-TCP or non-UDP packet and you specified a source or destination port, the SSR does not check the port value. The SSR checks only the source and destination IP addresses in the packet. You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 and 20 inclusive), >1024 (greater than 1024), <1024 (less than 1024), !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**.
<*DstPort*>	For TCP or UDP, the number of the destination TCP or UDP port. This field applies only to incoming TCP or UDP traffic. The same requirements and restrictions for <*SrcPort*> apply to <*DstPort*>.
<*tos*>	IP TOS (Type of Service) value. You can specify a TOS value from 0 – 15.
accounting	Valid with the **permit** command only. This keyword causes LFAP accounting information to be sent to the configured server for flows that match the ACL.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to permit IP traffic from the subnet 10.1.0.0 (with a 16 bit netmask) to any destination:

```
ssr(config)# acl 100 permit ip 10.1.0.0/16 any
```

The following command creates an ACL to deny any incoming TCP or UDP traffic coming from a privileged port (less than 1024). If the incoming traffic is not TCP or UDP, then the

SSR check only the source and destination addresses, not the port number. Therefore, this ACL will deny all non-TCP and non-UDP traffic.

```
ssr(config)# acl 120 deny ip any any 1-1024 any
```

To create an ACL to permit Telnet traffic (port 23) from the host 10.23.4.8 to the subnet 10.2.3.0:

```
ssr(config)# acl 130 permit ip 10.23.4.8 10.2.3.0/24
```

The following command creates an ACL to permit all IP traffic. Since none of the ACL fields are specified, they are all assumed to be wildcards.

```
ssr(config)# acl allip permit ip
```

The above command is equivalent to the following:

```
ssr(config)# acl allip permit ip any any any any any
```

acl permit|deny ip-protocol

Purpose

Create an ACL for any IP protocol type.

Format

acl <*name***> permit | deny ip-protocol** <*proto-num*> <*SrcAddr/Mask*> <*DstAddr/Mask*> <*tos*>

Mode

Configure

Description

The **acl permit ip-protocol** and **acl deny ip-protocol** commands define an Access Control List to allow or block IP traffic from entering or leaving the router for any protocol type. Unlike the more specific variants of the acl commands such as **ip**, **tcp** and **udp**, the **ip-protocol** version of the command allows the user to specify any valid IP protocol type. This command allows the user to specify an IP protocol other than the ones available with other **acl permit | deny** commands. For example, to specify an ACL for IP encapsulation in IP, one can use the IPinIP protocol type, 4, in the ACL. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the SSR assumes that the value is a wildcard (as if you had specified the **any** keyword).

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*proto-num*>	IP protocol number of this flow.
<*SrcAddr/Mask*>	The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

<*DstAddr/Mask*>	The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.
<*tos*>	IP TOS (Type of Service) value. You can specify a TOS from 0 – 15.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to permit VRRP traffic (IP protocol type 112) from the subnet 10.14.0.0 (with a 16 bit netmask) to any destination:

```
ssr(config)# acl 100 permit ip-protocol 112 10.14.0.0/16 any
```

The following command has the same function as **acl 120 deny igmp** since the protocol type for IGMP is 2.

```
ssr(config)# acl 120 deny ip-protocol 2
```

acl permit|deny ipx

Purpose

Create an IPX ACL.

Format

acl <*name*> **permit | deny ipx** <*SrcAddr*> <*SrcSocket*> <*DstAddr*> <*DstSocket*> <*SrcNetMask*> <*DstNetMask*>

Mode

Configure

Description

The **acl permit ipx** and **acl deny ipx** commands define an ACL to allow or block IPX traffic from entering or leaving the SSR.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*SrcAddr*>	The source IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition. To specify any network, enter **FFFFFFFF**.<*node*>; to specify any node, enter <*network*>.**FF:FF:FF:FF:FF:FF**.
<*SrcSocket*>	Source IPX socket. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.
<*DstAddr*>	The destination IPX address in <*network*>**.**<*node*> format. The syntax for the destination address is the same as the syntax for the source address <*SrcAddr*>. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.

<*DstSocket*> Destination IPX socket. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.

<*SrcNetmask*> Source network mask. This field specifies a group of networks for which the ACL applies. This mask field is ANDed with the network portion of <*SrcAddr*> and the source network of the incoming packets to determine a hit. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix.

This is an optional argument and if you omit the argument, the SSR uses the hexadecimal value FFFFFFFF.

<*DstNetmask*> Destination network mask. This field specifies a group of networks for which the ACL applies. This mask field is ANDed with the network portion of <*DstAddr*> and the destination network of the incoming packets to determine a hit. The SSR will interpret this number in hexadecimal format. You do not need to use a "0x" prefix.

This is an optional argument and if you omit the argument, the SSR uses the hexadecimal value FFFFFFFF.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

The following command creates an ACL to permit IPX traffic from the host with IPX address AAAAAAAA.01:20:0A:F3:24:6D, any socket, to any other IPX address (network.node), any socket.

```
ssr(config)# acl 100 permit ipx AAAAAAAA.01:20:0A:F3:24:6D any any any
```

The following command creates an ACL to deny IPX traffic from the host with IPX address F6D5E4.01:20:0A:F3:24:6D, with socket address 451, to any other IPX address (network.node), any socket.

```
ssr(config)# acl 200 deny ipx F6D5E4.01:20:0A:F3:24:6D 451 any any
```

acl permit|deny ipxgns

Purpose

Create an IPX GNS (Get Nearest Server) ACL.

Format

acl <*name*> **permit | deny ipxgns** <*ServerAddr*> <*ServiceType*> <*ServiceName*>

Mode

Configure

Description

The **acl permit ipxgns** and **acl deny ipxgns** commands define an ACL to allow or block replying to GNS requests.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*ServerAddr*>	The SAP server's IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. You can use the keyword **any** to specify a wildcard ("don't care") condition.
<*ServiceType*>	The SAP service type. Express the service type in hexadecimal. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.
<*ServiceName*>	The SAP service name. This is an optional argument and if you omit the argument, the SSR applies a wildcard condition to the field.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit

all traffic. You can only apply the **acl permit ipxgns** and **acl deny ipxgns** commands to output.

Examples

To create a GNS ACL to permit the SSR to reply with the server "FILESERVER", whose IPX address is F6D5E4.01:20:0A:F3:24:5D, to get nearest server requests:

```
ssr(config)# acl 100 permit ipxgns F6D5E4.01:20:0A:F3:24:5D 0004 FILESERVER
```

To create a GNS ACL to prevent the SSR from replying with the server "ARCHIVESERVER", whose IPX address is F6D5E4.01:20:0A:F3:24:5C, to a get nearest server request:

```
ssr(config)# acl 200 deny ipxgns F6D5E4.01:20:0A:F3:24:5C 0009 ARCHIVESERVER
```

acl permit|deny ipxrip

Purpose

Create an IPX RIP (Route Information Protocol) ACL.

Format

acl <*name*> **permit | deny ipxrip** <*FromNetwork*> <*ToNetwork*>

Mode

Configure

Description

The **acl permit ipxrip** and **acl deny ipxrip** commands define an ACL to allow or block IPX RIP traffic from entering or leaving the SSR.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*FromNetwork*>	The "from" IPX network address. You can use the **any** keyword to specify a wildcard condition. If you use **any**, the SSR uses the value 0 for <*FromNetwork*> *and FFFFFFFE for* <*ToNetwork*>.
<*ToNetwork*>	The "to" IPX network address. This is an optional parameter. If you omit this parameter, the value that the SSR assumes depends on whether you specified **any** for <*FromNetwork*>. –If you omit the <*ToNetwork*> value and you used the value **any** for <*FromNetwork*>, the SSR sets the <*ToNetwork*> to FFFFFFFE. –If you omit the <*ToNetwork*> value but do not use the value **any** for <*FromNetwork*>, the SSR sets <*ToNetwork*> to the same value you specified for <*FromNetwork*>.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to permit IPX RIP traffic from networks AA000001 to AAFFFFFF:

```
ssr(config)# acl 100 permit ipxrip AA000001 AAFFFFFF
```

acl permit|deny ipxsap

Purpose

Create an IPX SAP (Service Advertisement Protocol) ACL.

Format

acl <*name*> **permit | deny ipxsap** <*ServerAddr*> <*ServiceType*> <*ServiceName*>

Mode

Configure

Description

The **acl permit ipxsap** and **acl deny ipxsap** commands define an ACL to allow or block IPX SAP traffic from entering or leaving the SSR.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*ServerAddr*>	The SAP server's IPX address in <*network*>**.**<*node*> format, where <*network*> *is the network address and* <*node*> is the MAC address. You can use the keyword **any** to specify a wildcard ("don't care") condition. To specify any network, enter **FFFFFFFF**.<*node*>; to specify any node, enter <*network*>.**FF:FF:FF:FF:FF:FF**.
<*ServiceType*>	The SAP service type. Express the service type in hexadecimal. You do not need to use a "0x" prefix. You can use the keyword **any** to specify a wildcard ("don't care") condition.
<*ServiceName*>	The SAP service name. This is an optional argument and if you omit the argument, the SSR applies a wildcard condition to the field.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create a SAP ACL to permit SAP information related to the server "FILESERVER" whose IPX address is F6D5E4.01:20:0A:F3:24:5D:

```
ssr(config)# acl 100 permit ipxsap F6D5E4.01:20:0A:F3:24:5D 0004 FILESERVER
```

To create a SAP ACL to deny SAP information related to the server "ARCHIVESERVER" whose IPX address is F6D5E4.01:20:0A:F3:24:5C:

```
ssr(config)# acl 200 deny ipxsap F6D5E4.01:20:0A:F3:24:5C 0009 ARCHIVESERVER
```

acl permit|deny ipxtype20

Purpose

Create an IPX type 20 ACL.

Format

acl <*name*> **permit | deny ipxtype20**

Mode

Configure

Description

The **acl permit ipxtype20** and **acl deny ipxtype20** commands define an ACL to allow or block IPX type 20 packets from entering or leaving the SSR.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to deny IPX type 20 packets:

```
ssr(config)# acl 100 deny ipxtype20
```

acl permit|deny tcp

Purpose

Create a TCP ACL.

Format

acl <*name*> **permit | deny tcp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> [**accounting**][**established**]

Mode

Configure

Description

The **acl permit tcp** and **acl deny tcp** commands define an ACL to allow or block TCP traffic from entering or leaving the SSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the SSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword. The two exceptions to this rule are the optional parameters <*tos*> (type of service) and **accounting**. <*tos*> is a value from 0 to 15. The **accounting** keyword is only valid for the **permit** command, and can be placed anywhere on the command line. When you specify the **accounting** keyword, LFAP accounting information will be sent to the configured server for flows that match the ACL.

Parameters

<*name*>	Is the name of this ACL. You can use a string of characters or a number.
<*SrcAddr/Mask*>	Is the source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").
<*DstAddr/Mask*>	Is the destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

<SrcPort>	For TCP or UDP, is the number of the source TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 and 20 inclusive), >1024 (greater than 1024), <1024 (less than 1024), !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**.
<DstPort>	For TCP or UDP, is the number of the destination TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* The same requirements and restrictions for *<SrcPort>* apply to *<DstPort>*.
<tos>	Is the IP TOS (Type of Service) value. You can specify a TOS value from 0 – 15.
accounting	Is valid with the **permit** command only. This keyword causes LFAP accounting information to be sent to the configured server for flows that match the ACL.
established	Allows TCP responses from external hosts, provided the connection was established internally.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

To create an ACL to permit TCP traffic from the subnet 10.21.33.0 (with a 24 bit netmask) to any destination:

```
ssr(config)# acl 100 permit tcp 10.21.33.0/255.255.255.0 any
```

To create an ACL to deny any incoming HTTP traffic:

```
ssr(config)# acl noweb deny tcp any any http any
```

To create an ACL to permit FTP traffic (both command and data ports) from subnet 10.31.34.0 to 10.31.60.0:

```
ssr(config)# acl ftp100 permit tcp 10.31.34.0/24 10.31.60.0/24 20-21 any
```

acl permit|deny udp

Purpose

Create a UDP ACL.

Format

acl <*name*> **permit | deny udp** <*SrcAddr/Mask*> <*DstAddr/Mask*> <*SrcPort*> <*DstPort*> <*tos*> [**accounting**]

Mode

Configure

Description

The **acl permit udp** and **acl deny udp** commands define an ACL to allow or block UDP traffic from entering or leaving the SSR. For each of the values describing a flow, you can use the keyword **any** to specify a *wildcard* ("don't care") condition. If you do not specify a value for a field, the SSR applies a wildcard condition to the field, giving the same effect as if you specify the **any** keyword. The two exceptions to this rule are the optional parameters <*tos*> (type of service) and **accounting**. <*tos*> is a value from 0 to 15. The **accounting** keyword is only valid for the **permit** command, and can be placed anywhere on the command line. When you specify the **accounting** keyword, LFAP accounting information will be sent to the configured server for flows that match the ACL.

Parameters

<*name*>	Name of this ACL. You can use a string of characters or a number.
<*SrcAddr/Mask*>	The source address and the filtering mask of this flow. If the source address is a network or subnet address, you must supply the filtering mask. Generally, the filtering mask is the network mask of this network or subnet. If the source address is that of a host then no mask is required. By default, if a mask is not supplied, the source address is treated as that of a host. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").
<*DstAddr/Mask*>	The destination address and the filtering mask of this flow. The same requirements and restrictions for <*SrcAddr/Mask*> apply to <*DstAddr/Mask*>.

<SrcPort>	For TCP or UDP, the number of the source TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* You can specify a range of port numbers using operator symbols; for example, 10-20 (between 10 and 20 inclusive), >1024 (greater than 1024), <1024 (les than 1024), !=1024 (not equal to 1024). The port numbers of some popular services are already defined as keywords. For example, for Telnet, you can enter the port number 23 as well as the keyword **telnet**.
<DstPort>	For TCP or UDP, the number of the destination TCP or UDP port. *This field applies only to incoming TCP or UDP traffic.* The same requirements and restrictions for *<SrcPort>* apply to *<DstPort>*.
<tos>	IP TOS (Type of Service) value. You can specify a TOS value from 0 – 15.
accounting	Valid with the **permit** command only. This keyword causes LFAP accounting information to be sent to the configured server for flows that match the ACL.

Restrictions

When you apply an ACL to an interface, the SSR appends an *implicit deny rule* to that ACL. The implicit deny rule denies ***all*** traffic. If you intend to allow all traffic that doesn't match your specified ACL rules to go through, you must *explicitly* define a rule to permit all traffic.

Examples

Here are some examples of ACL commands for permitting and denying UDP traffic flows.

```
ssr(config)# acl 100 permit udp 10.1.3.0/24 any
```

Creates an ACL to permit UDP traffic from the subnet 10.1.3.0 (with a 24 bit netmask) to any destination.

```
ssr(config)# acl notftp deny udp any any tftp any
```

Creates an ACL to deny any incoming TFTP traffic.

```
ssr(config)# acl udpnfs permit udp 10.12.0.0/16 10.7.0.0/16 any nfs
```

Creates an ACL to permit UDP based NFS traffic from subnet 10.12.0.0 to subnet 10.7.0.0.

acl-policy enable external

Purpose

Allow an external server to create and delete ACLs.

Format

acl-policy enable external

Mode

Configure

Description

The **acl-policy enable external** command allows ACLs to be configured by an external agent, such as the Policy Manager. If this command is in the active configuration, an external server can create, modify, and delete ACLs on the SSR. If this command is not in the active configuration, then ACLs can only be created, modified, and deleted using the CLI.

Parameters

None.

Restrictions

The only action allowed by the **acl-policy enable external** command is to allow an external server to create, modify, and delete ACLs. Once entered, this command must be negated in order to prohibit an external server from creating, altering, or deleting ACLs. An external server can only modify ACLs that it created, or ACLs that were created using the CLI with the “external” flag. It cannot modify an ACL that was created using the CLI with the “local” flag.

Chapter 2

acl-edit Commands

The acl-edit command activates the ACL Editor mode. The ACL Editor provides a user-friendly interface for maintaining and manipulating rules in an ACL. Using the editor, you can add, delete or re-order ACL rules. In addition, if the modified ACL is currently applied to an interface, the ACL is automatically "re-applied" to the interface and takes effect immediately. To edit an ACL, you enter the **acl-edit** command in Configure mode. The command must also specify the name of the ACL you want to edit. Only one ACL can be edited at one time.

Command Summary

Table 2 lists the commands available with the ACL Editor. The sections following the table describe the command syntax.

Table 2. acl-edit commands

acl-edit <*aclname*>
acl permit | **deny**
delete <*rule#*>
exit
move <*rule#*> **after** <*rule#*>
save
show

acl-edit

Purpose

Enter ACL Editor to edit the specified ACL.

Format

acl-edit <*aclname*>

Mode

Configure

Description

The **acl-edit** command enters the ACL Editor to edit an ACL specified by the user. Once inside the ACL editor, the user can then add, delete or re-order ACL rules for that ACL. If the ACL happens to be applied to an interface, changes made to that ACL will automatically take effect when the changes are committed to the running system.

Parameters

<*aclname*> Name of the ACL to edit.

Restrictions

Inside the ACL Editor, you can only add rules for the ACL you specified in the acl-edit command. You cannot add rules for other ACLs. Basically, each ACL editing session works only on one ACL at a time. For example, if you start with *acl-edit 110*, you cannot add rules for ACL *121*.

Example

To edit ACL 111:

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any

ssr(acl-edit)> ?
acl                     - Configure L3 Access Control List
delete                  - Delete an ACL rule
exit                    - Exit current mode
move                    - Move an ACL rule
save                    - Save changes made to this ACL
show                    - Show contents of this ACL
```

acl permit|deny

Purpose

Create an ACL rule to permit or deny traffic.

Format

acl <*name*> **permit | deny**

Mode

ACL Editor

Description

The **acl permit | deny** commands are equivalent to the same commands in the Configuration mode. You can use these commands to create rules for the ACL that you are editing. Just like the acl commands in Configuration mode, new rules are appended to the end of the rules. You can use the move command to re-order the rules.

Restrictions

You can only add rules for the ACL you specified in the acl-edit command. You cannot add rules for other ACLs. For example, if you start with *acl-edit 110*, you cannot add rules for ACL *121*.

Example

To add a new rule (deny all UDP traffic) to ACL 111:

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any

ssr(acl-edit)> acl 111 deny udp
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp
```

delete

Purpose

Deletes a rule from an ACL.

Format

delete <*rule#*>

Mode

ACL Editor

Description

The **delete** commands allows the administrator to delete a specific rule from an ACL. When in the ACL Editor, each rule is displayed with its rule number. One can delete a specific rule from an ACL by specifying its rule number with the delete command.

Parameters

<*rule#*> Number of the ACL rule to delete.

Restrictions

None

Example

To delete ACL rule number 2 from the ACL:

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

ssr(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp
```

exit

Purpose

Exit ACL Editor.

Format

exit

Mode

ACL Editor

Description

The **exit** command allows the user to exit the ACL Editor. Before exiting, if changes are made to this ACL, the system will prompt the user to see if the changes should be committed to the running system or discarded. If the user commits the changes then changes made to this ACL will take effect immediately. If the ACL is applied to an interface, the ACL is automatically re-applied to the interface. Packets going through this interface will be matched against the new rules in this ACL. If the user chooses not to commit the changes, the changes will be discarded. The next time the user edits this ACL, changes from the previous edit session will be lost.

Parameters

None

Restrictions

None

Example

To create an ACL to deny IGMP traffic from the subnet 10.1.5.0 (with a 24 bit netmask) to any destination:

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

ssr(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp

ssr(acl-edit)> exit

ssr(config)# acl 410 deny igmp 10.1.5.0/24 any
```

move

Purpose

Re-order ACL rules by moving a rule to another position.

Format

move <*src-rule#*> **after** <*dst-rule#*>

Mode

ACL Editor

Description

The **move** command provides the user with the ability to re-order rules within an ACL. When new rules are entered in the ACL Editor, they are appended to the end of the rules. One can move these rules to the desired location by using the move command. The move command can also be used on existing ACL rules created in Configuration mode instead of the ACL Editor.

Parameters

<*src-rule#*> Rule number of the rule you want to move.

<*dst-rule#*> Rule number of the rule after which you want the source rule to move to.

Restrictions

None

Examples

To move rule #2 to the end of the list:

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 permit udp 10.1.17.0/24 10.1.22.0/24 2000-2002 any
4*: acl 111 permit udp 10.1.18.0/24 10.1.34.0/24 2003-2005 any

ssr(acl-edit)> move 2 after 4
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit udp 10.1.17.0/24 10.1.22.0/24 2000-2002 any
3*: acl 111 permit udp 10.1.18.0/24 10.1.34.0/24 2003-2005 any
4*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
```

save

Purpose

Save any changes made by the ACL Editor.

Format

save

Mode

ACL Editor

Description

The **save** command saves any non-committed changes made by the ACL Editor. If changes are made to this ACL, the changes will be saved and will take effect immediately. If the ACL is applied to an interface, the ACL is automatically re-applied to the interface. Packets going through this interface will be matched against the new rules in this ACL. The **save** command also contains an implicit exit command. Regardless of whether changes were made by the ACL Editor or not, upon completion of the **save** command, the user exits the ACL Editor and returns to Configuration mode. Consequently, one should issue the **save** command after all the changes are made.

Parameters

None

Restrictions

None

Examples

To save and commit the changes made by the ACL Editor.

```
ssr(config)# acl-edit 111
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
3*: acl 111 deny udp

ssr(acl-edit)> delete 2
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 deny udp

ssr(acl-edit)> save
```

show

Purpose

Displays the contents of the ACL in the current editing session.

Format

show

Mode

ACL Editor

Description

The **show** command displays the contents of the ACL currently being edited.

Parameters

None

Restrictions

None

Examples

To display the contents of the ACL currently being edited:

```
ssr(acl-edit)# show
1*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2000-2002 any
2*: acl 111 permit tcp 10.1.15.0/24 10.1.11.0/24 2003-2005 any
```

Chapter 3

aging Commands

The **aging** commands control aging of learned MAC address entries in the SSR's L2 lookup tables or layer3/4 flows. Using the **aging** commands, you can show L2 or layer 3/4 aging information, set or disable L2 aging on specific ports, set or disable aging of layer 3/4 flows, or set or disable NAT or LSNAT flows.

Command Summary

Table 3 lists the **l2** and **l3** aging commands. The sections following the table describe the command syntax.

Table 3. aging commands

aging l2 disable <*port-list*> | **all-ports**
aging l2 set aging-timeout <*seconds*> **port** <*port-list*> | **all-ports**
aging l2 show status
aging l3 set timeout <*seconds*> | **disable**
aging l3 set nat-flow-timeout <*minutes*> | **disable**
aging l3 show status

aging l2 disable

Purpose

Disable aging of MAC addresses.

Format

aging l2 disable <*port-list*> | **all-ports**

Mode

Configure

Description

By default, the SSR ages learned MAC addresses in the L2 lookup tables. Each port has its own L2 lookup table. When a learned entry ages out, the SSR removes the aged out entry. You can disable this behavior by disabling aging on all ports or on specific ports.

Parameters

<*port-list*> | **all-ports**
: The port(s) on which you want to disable aging. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, aging is disabled on all ports.

Restrictions

Unknown.

Examples

To disable aging on slot 1, port 3:

```
ssr(config)# aging l2 disable et.1.3
```

To disable aging on slot 4, port 2, and slots 1 through 3, ports 4, 6, 7, and 8:

```
ssr(config)# aging l2 disable et.4.2 et.(1-3).(4 6-8)
```

To disable aging on all ports:

```
ssr(config)# aging l2 disable all-ports
```

aging l2 set aging-timeout

Purpose

Set the aging time for learned MAC entries.

Format

aging l2 set <*port-list*> | **all-ports aging-timeout** <*seconds*>

Mode

Configure

Description

The **aging l2 set aging-timeout** command sets the aging time for learned MAC entries. When the aging time expires for a MAC address, the SSR removes the MAC address from the specified port(s). The aging time is specified in seconds.

Parameters

<*port-list*> | **all-ports**
The port(s) on which you want to set the aging time. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, the aging time is set on all ports.

<*seconds*> The number of seconds the SSR allows a learned MAC address to remain in the L2 lookup table (for the specified port). You can specify from 15 to 1000000 seconds. The default is 300 seconds.

Restrictions

None.

Example

To set the aging time to 15 seconds on all ports:

```
ssr(config)# aging l2 set all-ports aging-timeout 15
```

aging l2 show status

Purpose

Show the L2 aging status for SSR ports.

Format

aging l2 show status

Mode

User

Description

The **aging l2 show status** command shows whether L2 aging is enabled or disabled on SSR ports. For ports on which L2 aging is enabled, this command also shows the aging time.

Parameters

None.

Restrictions

None.

aging l3 set timeout

Purpose

Set the aging time for a layer 3/4 flow.

Format

aging l3 set timeout <*seconds*> | **disable**

Mode

Configure

Description

The **aging l3 set timeout** command sets the aging time for a layer 3/4 flow. The aging time is specified in seconds.

Parameters

<*seconds*> The number of seconds the SSR allows for a layer 3/4 flow. You can specify a value from 30 to 3600 seconds. For example, in an ISP environment (where thousands of flows are possible), you could change this value to 180-300 (3-5 minutes) to help in keeping with longer-term flows. The default is 30 seconds.

disable Disables layer 3/4 aging.

Restrictions

None.

Example

To set the layer 3/4 flow aging time to 300 seconds (5 minutes):

```
ssr(config)# aging l3 set timeout 60
```

aging l3 set nat-flow-timeout

Purpose

Set the aging time for NAT and LSNAT flows.

Format

aging l3 set nat-flow-timeout <*minutes*> | **disable**

Mode

Configure

Description

The **aging l3 set nat-flow-timeout** command sets the aging time for Network Address Translation (NAT) and Load Sharing NAT flows. The aging time is specified in minutes.

Parameters

<*minutes*> The number of minutes the SSR allows for NAT and LSNAT flows. You can specify from 2 to 120 minutes. The default is 2 minutes.

disable Disables NAT and LSNAT flow aging.

Restrictions

None.

Example

To set the NAT aging time to 5 minutes:

```
ssr(config)# aging l3 set nat-flow-timeout 5
```

aging l3 show status

Purpose

Show the L3 aging status for SSR ports.

Format

aging l3 show status

Mode

User

Description

The **aging l3 show status** command shows whether layer 3/4 aging is enabled or disabled on SSR ports. For ports on which layer 3/4 aging is enabled, this command also shows the aging time.

Parameters

None.

Restrictions

None.

Example

To show whether layer 3/4 aging is enabled and the aging time for enabled ports:

```
ssr# aging l3 show status
L3 Aging: Timeout 30 seconds
```

Chapter 4

arp Commands

The **arp** commands enable you to add, display, and clear ARP entries on the SSR.

Command Summary

Table 4 lists the arp commands. The sections following the table describe the command syntax.

Table 4. arp commands

arp add <*host*> **mac-addr** <*MAC-addr*> **exit-port** <*port*> **keep-time** <*seconds*>
arp clear <*host*> | **all**
arp set interface <*name*> | **all keep-time** <*number*>
arp show <*IPaddr*> | **all**
statistics show arp

arp add

Purpose

Add an ARP entry.

Format

arp add <*host*> **mac-addr** <*MAC-addr*> **exit-port** <*port*> **keep-time** <*seconds*>

Mode

Enable and Configure

Description

The **arp add** command lets you manually add ARP entries to the ARP table. Typically, the SSR creates ARP entries dynamically. Using the **arp add** command, you can create an ARP entry to last a specific amount of time or as a permanent ARP entry. This command exists in both Enable and Configure mode with a slight variation. The **keep-time** option is valid only in Enable mode. The **keep-time** option allows you to create an ARP entry to last a specific amount of time. The Configure mode version of the **arp add** command does not use the **keep-time** option. ARP entries created in the Configure mode are permanent ARP entries and they do not have an expiration time. If the exit port is not specified, then packets to the IP address for which the ARP entry is created are transmitted on all ports of the interface. If an ARP request is received from the host for which the ARP entry was created, then the exit port is updated with the port on which the ARP request was received, so that subsequent packets are transmitted on one port only.

Parameters

<*host*>	Hostname or IP address of this ARP entry.
mac-addr <*MAC-addr*>	MAC address of the host.
exit-port <*port*>	The port for which you are adding the entry. Specify the port to which the host is connected.
keep-time <*seconds*>	The number of seconds this ARP entry should remain in the ARP table. A value of 0 means this is a permanent ARP entry.

Note: This option is valid only for the Enable mode **arp add** command.

Restrictions

If you enter the **arp add** command while in the Configure mode, you can add only permanent ARP entries.

Examples

To create an ARP entry for the IP address 10.8.1.2 at port et.4.7 for 15 seconds:

```
ssr# arp add 10.8.1.2 mac-addr 08:00:20:a2:f3:49 exit-port et.4.7 keep-time 15
```

To create a permanent ARP entry for the host *nfs2* at port et.3.1:

```
ssr(config)# arp add nfs2 mac-addr 080020:13a09f exit-port et.3.1
```

arp clear

Purpose

Remove an ARP entry from the ARP table.

Format

arp clear <*host*> | **all**

Mode

Enable

Description

The **arp clear** command lets you manually remove entries from the ARP table. The command can remove both dynamic and permanent entries.

Parameters

<*host*>	Hostname or IP address of the ARP entry to remove.
all	Remove all ARP entries, thus clearing the entire ARP table.

Examples

To remove the ARP entry for the host 10.8.1.2 from the ARP table:.

```
ssr# arp clear 10.8.1.2
```

To clear the entire ARP table.

```
ssr# arp clear all
```

If the Startup configuration file contains **arp add** commands, the Control Module re-adds the ARP entries even if you have cleared them using the **arp clear** command. To

permanently remove an ARP entry, use the **negate** command or **no** command to remove the entry. Here is an example of the **no** command:

```
ssr# no arp add nfs2 macaddr 080020:13a09f exit-port et.3.1
```

This command removes the ARP entry for "nfs2".

arp set interface

Purpose

Set the lifetime of ARP entries in seconds.

Format

arp set interface <*name*> | **all keep-time <*number*>**

Mode

Configure

Description

The **arp set interface ... keep-time** command lets you specify the lifespan (inseconds) for any or all ARP interface entries.

Parameters

interface <*name*> | **all**	Name of the interface(s) for which you will define the lifespan.
keep-time <*number*>	number of seconds determining lifespan of ARP interfaces. The default value is 1200 seconds (20 minutes).

arp show

Purpose

Display the ARP table.

Format

arp show <*IPaddr*> | **all**

Mode

Enable

Description

The **arp show** command displays the entire ARP table.

Parameters

<*IPaddr*>	Shows the ARP entry for the specified IP address.
all	Shows all entries in the ARP table.

statistics show arp

Purpose

Display ARP statistics.

Format

statistics show arp <*Interface Name*> | **all**

Mode

Enable

Description

The **arp show statistics** command displays ARP statistics, such as the total number of ARP requests and replies.

Parameters

<*Interface Name*>	Displays ARP statistics for the specified interface.
all	Displays ARP statistics for all router interfaces.

Chapter 5

bgp Commands

The **bgp** commands let you display and set parameters for the Border Gateway Protocol (BGP).

Command Summary

Table 5 lists the **bgp** commands. The sections following the table describe the command syntax.

Table 5. bgp commands

bgp add network <*ipaddr-mask*> | **all group** <*number-or-string*>
bgp add peer-host <*ipaddr*> **group** <*number-or-string*>
bgp clear peer-host _ipaddr>
bgp create peer-group <*number-or-string*>
bgp set DampenFlap <*option*>
bgp set default-metric <*num*>
bgp set cluster-id <*ipaddr*>
bgp set peer-group <*number-or-string*>
bgp set peer-host <*ipaddr*>
bgp set preference <*num*>
bgp show aspaths <*aspath*> | **all [to-terminal | to-file]**
bgp show cidr-only <*ip-addr-mask*> | **default | all [to-terminal | to-file]**

Table 5. bgp commands (Continued)

bgp show community community-id <*number*> **autonomous-system** <*number*> **| well-known-community [no-export | no-advertise | no-export-subconfed] | reserved-community** <*number*>**] [to-terminal | to-file]**
bgp show peer-as <*number*> **[to-terminal | to-file]**
bgp show peer-group-type external | internal | igp | routing [to-terminal | to-file]
bgp show peer-host <*ipaddr*> **received-routes | all-received-routes | advertised-routes [to-terminal | to-file]**
bgp show routes <*ip-addr-mask*> **| default | all [to-terminal | to-file]**
bgp show summary [to-terminal | to-file]
bgp show sync-tree
bgp start | stop
bgp trace <*option*>

bgp add network

Purpose

Adds a network to a BGP peer group.

Format

bgp add network <*ip-addr-mask*> | **all group** <*number-or-string*>

Mode

Configure

Description

The **bgp add network** command lets you add a BGP peer network, thus allowing peer connections from any addresses in the specified range of network and mask pairs.

Parameters

network <*ip-addr-mask*> | **all**
Specifies a network from which peer connections are allowed. Specify an IP address and Mask value. Example: 1.2.3.4/255.255.0.0 or 1.2.3.4/16. Specify **all** to add all networks.

group <*number-or-string*>
Specifies the group ID associated with this network range.

Restrictions

None.

bgp add peer-host

Purpose

Add a BGP peer by adding a peer host.

Format

bgp add peer-host <*ipaddr*> **group** <*number-or-string*>

Mode

Configure

Description

The **bgp add peer-host** command adds a peer-host to a BGP group.

Parameters

peer-host <*ipaddr*>
Specifies the peer host's IP address.

group <*number-or-string*>
Specifies the group ID of the group to which the peer host belongs.

Restrictions

None.

bgp clear peer-host

Purpose

Removes a BGP peer host.

Format

bgp clear peer-host <*ipaddr*>

Mode

Configure

Description

The **bgp clear peer-host** command removes a peer-host from a BGP group.

Parameters

peer-host <*ipaddr*>
Specifies the peer host's IP address.

Restrictions

None.

bgp create peer-group

Purpose

Create a BGP Group based on type or the autonomous system of the peers. You can create any number of groups, but each group must have a unique combination of type and peer autonomous system.

Format

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]**

Mode

Configure

Description

The **bgp create peer-group** command creates a BGP peer group.

Parameters

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. 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.

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.

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.

autonomous-system
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>*
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.

Restrictions

None.

bgp set cluster-id

Purpose

Specifies the route reflection cluster ID for BGP.

Format

bgp set cluster-id <*ipaddr*>

Mode

Configure

Description

The **bgp set cluster-id** command specifies the route reflection cluster ID for BGP. The cluster ID defaults to the same as the router-id. If a router is to be a route reflector, then a single cluster ID should be selected and configured on all route reflectors in the cluster. If there is only one route reflector in the cluster, the cluster ID setting may be omitted, as the default will suffice.

Parameters

cluster-id <*ipaddr*>
Is the cluster ID.

Restrictions

The only constraints on the choice of cluster ID are (a) IDs of clusters within an AS must be unique within that AS, and (b) the cluster ID must not be 0.0.0.0. Choosing the cluster ID to be the router ID of one router in the cluster will always fulfill these criteria.

bgp set peer-group

Purpose

Set parameters for the specified BGP Peer Group.

Format

bgp set peer-group <*number-or-string*> **[med | reflector-client | no-client-reflect | [metric-out** <*num*>**] | [set-pref** <*num*>**][local-as** <*num*>**] | ignore-first-as-hop | [generate-default enabled | disabled] | [gateway** <*ipaddr*>**] | next-hop-self | [preference** <*num*>**][preference2** <*num*>**][local-address** <*ipaddr*>**] | [hold-time** <*num*>**] | [version 2 | 3 | 4] | passive | [send-buffer** <*num*>**] | [recv-buffer** <*num*>**] | [in-delay** <*num*>**] | [out-delay** <*num*>**] | [keep all | none] | show-warnings | no-aggregator-id | keep-alives-always | v3-asloop-okay | no-v4-asloop | [as-count** <*num*>**] | log-up-down | [ttl** <*num*>**] | [optional-attributes-list** <*number-or-string*>**]]**

Mode

Configure

Description

The **bgp set peer-group** command sets parameters for the specified BGP group.

Parameters

group <*number-or-string*>
Specifies the group.

med
Forces med to be used for route selection process. By default, any metric (Multi_Exit_Disc, or MED) received on a BGP connection is ignored. If it is desired to use MEDs in route selections, the **med** option must be specified in this (**create peer-group**) command. By default, MEDs are not sent on external connections. To send MEDs, use the **metric** option of the **create bgp-export-destination** statement or the **metric-out** option of the **set peer-group** or **set peer-host** commands.

reflector-client
The **reflector-client** option specifies that GateD will act as a route reflector for this group. All routes received from any group member will be sent to all other internal

neighbors, and all routes received from any other internal neighbors will be sent to the reflector clients. Since the route reflector forwards routes in this way, the reflector-client group need not be fully meshed. Use only for INTERNAL, ROUTING and IGP groups.

no-client-reflect
If the no-client-reflect option is specified, routes received from reflector clients will only be sent to internal neighbors which are not in the same group as the sending reflector client. In this case the reflector-client group should be fully meshed. In all cases, routes received from normal internal peers will be sent to all reflector clients.

Note that it is necessary to export routes from the local AS into the local AS when acting as a route reflector. The reflector-client option specifies that GateD will act as a route reflector for this group. All routes received from any group member will be sent to all other internal neighbors, and all routes received from any other internal neighbors will be sent to the reflector clients. Since the route reflector forwards routes in this way, the reflector-client group need not be fully meshed.

metric-out <*num*>
Specifies the primary metric used on all routes sent to the specified peer group. Specify a number from 0 - 65535.

set-pref <*num*>
Routes propagated by IBGP must include a Local_Pref attribute. By default, BGP sends the Local_Pref path attribute as 100, and ignores it on receipt. GateD BGP does not use Local_Pref as a route-preference decision maker unless the setpref option has been set. For Routing- or Internal-type groups, the setpref option allows GateD's global protocol preference to be exported into Local_Pref and allows Local_Pref to be used for GateD's route selection preference. Note that the setpref option is the only way for GateD to send a route with a given local_pref. The local_pref is never set directly, but rather as a function of the GateD preference and setpref metrics. Allows BGP's LOCAL_PREF attribute to be used to set the GateD preference on reception, and allows the GateD preference to set the LOCAL_PREF on transmission. The set-pref metric works as a lower limit, below which the imported LOCAL_PREF may not set the GateD preference. Use only for INTERNAL, ROUTING, and IGP groups. Specify a number from 0 - 255.

local-as <*num*>
Identifies the autonomous system which the router is representing to this group of peers. The default is the one configured by the **set autonomous_system** command. Specify a number from 1 - 65534.

ignore-first-as-hop
Some routers, known as Route Servers, are capable of propagating routes without appending their own AS to the AS path. By default, GateD will drop such routes. Specifying ignore-first-as-hop here or on either the **create peer-group** or **set peer-host** CLI commands disables this feature. This option should only be used if it is positively known that the peer is a route server and not a normal router.

generate-default enabled | disabled
Specifies whether the router should generate a default route when BGP receives a valid update from its peer. If this option is not specified, then the generation of default route is enabled.

gateway <*ipaddr*>
If a network is not shared with a peer, this option specifies a router on an attached network to be used as the next hop router for routes received from this neighbor. This field is used for EBGP Multihop. **The IP address must be a host address on a locally attached network**.

next-hop-self
This option causes the next hop in route advertisements set to this peer or group of peers to be set to our own router's address even if it would normally be possible to send a third-party next hop. Use of this option may cause efficient routes to be followed, but it may be needed in some cases to deal with broken bridged interconnect media (in cases where the routers on the shared medium do not really have full connectivity to each other) or broken political situations. Use only for EXTERNAL groups.

preference <*num*>
Specifies the preference used for routes learned from these peers. Specify a number from 0 - 255.

preference2 <*num*>
In case of a preference tie, this option (the second preference), may be used to break the tie. The default value is 0. Specify a number from 0 - 255.

local-address <*ipaddr*>
Specifies the address to be used on the local end of the TCP connection with the peer or with the peer's gateway when the gateway option is used. A session with an external peer will only be opened when an interface with the appropriate local address (through which the peer or gateway address is directly reachable). In either case incoming connections will only be recognized as matching a configured peer if they are addressed to the configured local address. Use only for INTERNAL, ROUTING, and IGP groups. **It should be one of the interface addresses.**

hold-time <*num*>
Specifies the hold time value to use when negotiating the connection with this peer, in seconds. If BGP does not receive a keepalive, update, or notification message from a peer within the period specified in the Hold Time field of the BGP Open message, then the BGP connection will be closed. The value must be either 0 (no keepalives will be sent) or at least 6.

version 2 | 3 | 4
Specifies the version of the BGP protocol to use with this peer. If not specified, only the specified version will be offered. Specify 2, 3, or 4.

passive
Specifies that active OPENs to this peer should not be attempted. BGP would wait for

the peer to issue an OPEN. By default, all explicitly configured peers are active, they periodically send OPEN messages until the peer responds. Note that if it is applied to both sides of a peering session, it will prevent the session from ever being established.

send-buffer <*num*>
Controls the amount of send buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

recv-buffer <*num*>
Controls the amount of receive buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

in-delay <*num*>
Used to dampen route fluctuations. In delay specifies the amount of time in secs a route learned from a BGP peer must be stable before it is accepted into the routing database. Specify a number equal to or greater than 0. The default value is 0, meaning that this feature is disabled.

out-delay <*num*>
Used to dampen route fluctuations. Out delay is the amount of time in secs a route must be present in the routing table before it is exported to BGP. Specify a number equal to or greater than 0. The default value is 0, meaning that this feature is disabled.

keep all | none
Used to retain routes learned from a peer even if the routes' AS paths contain one of our exported AS numbers.

show-warnings
This option causes GateD to issue warning messages when receiving questionable BGP updates such as duplicate routes and/or deletions of non-existing routes. Normally these events are silently ignored.

no-aggregator-id
This option causes GateD to specify the router ID in the aggregator attribute as zero (instead of its router ID) in order to prevent different routers in an AS from creating aggregate routes with different AS paths.

keep-alives-always
This option causes GateD to always send keepalives, even when an update could have correctly substituted for one. This allows interoperability with routers that do not completely obey the protocol specifications on this point.

v3-asloop-okay
By default GateD will not advertise routes whose AS path is looped (i.e. with an AS appearing more than once in the path) to version 3 external peers. Setting this flag removes this constraint. Ignored when set on internal groups or peers.

no-v4-asloop
Prevents routes with looped AS paths from being advertised to version 4 external peers. This can be useful to avoid advertising such routes to peer which would incorrectly forward the routes on to version 3 neighbors.

as-count <*num*>
This option determines how many times the SSR will insert its own AS number when we send the AS path to an external neighbor.

Specify a number between 1 and 25. The default is 1. Higher values typically are used to bias upstream neighbors' route selection. (All else being equal, most routers will prefer to use routes with shorter AS Paths. Using **ascount**, the AS Path the SSR sends can be artificially lengthened.)

Note that ascount supersedes the **no-v4-asloop** option–regardless of whether **no-v4-asloop** is set, we will still send multiple copies of our own AS if the **as-count** option is set to something greater than one. Also, note that if the value of ascount is changed and GateD is reconfigured, routes will not be sent to reflect the new setting. If this is desired, it will be necessary to restart the peer session.

log-up-down
This option causes a message to be logged via the SYSLOG mechanism whenever a BGP peer enters or leaves the ESTABLISHED state.

ttl <*num*>
By default, BGP sets the IP TTL for local peers to ONE and the TTL for non-local peers to 255. This option is provided when attempting to communicate with improperly functioning routers that ignore packets sent with a TTL of ONE. Specify a number between 1 and 255.

optional-attributes-list <*number-or-string*>
Specifies the ID of the optional-attributes-list to be associated with this peer-group.

Restrictions

None.

bgp set DampenFlap

Purpose

Configures parameters for Weighted Route Dampening.

Format

bgp set dampenflap [state enable | disable] | [suppress-above <*num*>**] | [reuse-below** <*num*>**] | [max-flap** <*num*>**] | [unreach-decay** <*num*>**] | [reach-decay** <*num*>**] | [keep-history** <*num*>**]**

Mode

Configure

Description

The **bgp set dampenflap** command configures the state of Weighted Route Dampening.

Parameters

state enable | disable
Causes the Route Instability History to be maintained (**enable** option) or not (**disable** option).

suppress-above <*num*>
Is the value of the instability metric at which route suppression will take place. A route will not be installed in the FIB or announced even if it is reachable during the period that it is suppressed. The default is 3.0.

reuse-below <*num*>
Is the value of the instability metric at which a suppressed route will become unsuppressed, if it is reachable but currently suppressed. The value must be less than that for the suppress-above option. The default is 2.0.

max-flap <*num*>
Is the upper limit of the instability metric. This value must be greater than the larger of 1 and that for suppress-above. The default is 16.0.

unreach-decay <*num*>
Specifies the time in seconds for the instability metric value to reach one-half of its

current value when the route is *unreachable*. This half-life value determines the rate at which the metric value is decayed. The default is 900.

reach-decay <*num*>
Specifies the time in seconds for the instability metric value to reach one half of its current value when the route is *reachable*. This half-life value determines the rate at which the metric value is decayed. A smaller half-life value will make a suppressed route reusable sooner than a larger value. The default is 300.

keep-history <*num*>
Specifies the period in seconds over which the route flapping history is to maintained for a given route. The size of the configuration arrays is directly affected by this value. The default is 1800.

Restrictions

None.

bgp set default-metric

Purpose

Set the metric used when advertising routes through BGP.

Format

bgp set default-metric <*num*>

Mode

Configure

Description

The **bgp set default-metric** command lets you set the default metric BGP uses when it advertises routes. If this command is not specified, no metric is propagated. This metric may be overidden by a metric specified on the neighbor or group statements or in an export policy.

Parameters

<*num*> Specifies the default cost. Specify a number from 0 - 65535.

Restrictions

None.

bgp set peer-host

Purpose

Set parameters for a BGP peer host.

Format

bgp set peer-host <*ipaddr*> **[group** <*number-or-string*> | **[metric-out** <*num*>**]** | **[set-pref** <*num*>**][local-as** <*num*>**]** | **ignore-first-as-hop** | **[generate-default enabled** | **disabled]** | **[gateway** <*ipaddr*>**]** | **next-hop-self** | **[preference** <*num*>**][preference2** <*num*>**][local-address** <*ipaddr*>**]** | **[hold-time** <*num*>**]** | **[version 2** | **3** | **4]** | **passive** | **[send-buffer** <*num*>**]** | **[recv-buffer** <*num*>**]** | **[in-delay** <*num*>**]** | **[out-delay** <*num*>**]** | **[keep all** | **none]** | **show-warnings** | **no-aggregator-id** | **keep-alives-always** | **v3-asloop-okay** | **no-v4-asloop** | **[as-count** <*num*>**]** | **[ttl** <*num*>**]** | **[optional-attributes-list** <*number-or-string*>**]]**

Mode

Configure

Description

The **bgp set peer-host** command lets you set various parameters for the specified BGP peer hosts.

Parameters

group <*number-or-string*>
Specifies the group ID

metric-out <*num*>
Specifies the primary metric used on all routes sent to the specified peer group. The metric hierarchy is as follows, starting from the most preferred: 1)The metric specified by export policy. 2) Peer-level metricout. 3) Group-level metricout 4) Default metric. For INTERNAL, IGP, and ROUTING hosts use the **group** command to set the metric-out. Specify a number from 0 - 65535.

set-pref <*num*>
Allows BGP's LOCAL_PREF attribute to be used to set the GateD preference on reception, and allows the GateD preference to set the LOCAL_PREF on transmission.

The set-pref metric works as a lower limit, below which the imported LOCAL_PREF may not set the GateD preference. For INTERNAL, IGP, and ROUTING hosts, use the **group** command to set the metric-out. Specify a number from 0 - 255. **This parameter applies only to INTERNAL, IGP, and ROUTING hosts only.**

local-as <*num*>
Identifies the autonomous system which the router is representing to this group of peers. The default is the one configured using the **set autonomous_system** command. Specify a number from 1 - 65534.

ignore-first-as-hop
Some routers, known as Route Servers, are capable of propagating routes without appending their own AS to the AS path. By default, GateD will drop such routes. Specifying ignore-first-as-hop here or on either the **create peer-group** or **set peer-host** CLI commands disables this feature. This option should only be used if it is positively known that the peer is a route server and not a normal router.

generate-default enabled | disabled
Specifies whether the router should generate a default route when BGP receives a valid update from its peer. If this option is not specified, then the generation of default route is enabled.

gateway <*IPaddr*>
if a network is not shared with a peer, this option specifies a router on an attached network to be used as the next hop router for routes received from this neighbor. This is used for **EBGP multihop**. **The IP address must be a host address on a locally attached network.**

next-hop-self
This option causes the next hop in route advertisements set to this peer or group of peers to be set to our own router's address, even if it would normally be possible to send a third-party next hop. Use of this option may cause inefficient routes to be followed, but it may be needed in some cases to deal with broken bridged interconnect media (in cases where the routers in the shared medium do not really have full connectivity to each other) or broken political situations. **Use only for external peer hosts.**

preference <*num*>
Specifies the preference used for routes learned from these peers. This can differ from the default BGP preference set in the bgp set preference statement, so that GateD can prefer routes from one peer, or group of peer, over others. This preference may be explicitly overridden by import policy. Specify a number from 0 - 255.

preference2 <*num*>
In case of preference tie, this option (the second preference), may be used to break the tie. The default value is 0. Specify a number from 0 - 255.

local-address <*IPaddr*>
Specifies the address to be used on the local end of the TCP connection with the peer or with the peer's gateway when the gateway option is used. A session with an

external peer will only be opened when an interface with the appropriate local address (through which the peer or gateway address is directly reachable). In either case incoming connections will only be recognized as matching a configured peer if they are addressed to the configured local address. For INTERNAL, IGP and ROUTING, hosts use the **group** command to set the local-address. **It should be one of the interface addresses.**

hold-time <*num*>
Specifies the hold time value to use when negotiating the connection with this peer, in seconds. If BGP does not receive a keepalive, update, or notification message from a peer within the period specified in the Hold Time field of the BGP Open message, then the BGP connection will be closed. The value must be either 0 (no keepalives will be sent) or at least 6.

version 2 | 3 | 4
Specifies the version of the BGP protocol to use with this peer. If not specified, only the specified version will be offered. Specify 2, 3, or 4.

passive
Specifies that active OPENs to this peer should not be attempted. BGP would wait for the peer to issue an OPEN. By default, all explicitly configured peers are active, they periodically send OPEN messages until the peer responds. Note that if it is applied to both sides of a peering session, it will prevent the session from ever being established.

send-buffer <*num*>
Controls the amount of send buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 - 65535.

recv-buffer <*num*>
Controls the amount of receive buffer acquired from the memory subsystem. The maximum supported is 65535 bytes. By default, BGP acquires the maximum supported. Specify a number from 4096 – 65535.

in-delay <*num*>
Used to dampen route fluctuations. In delay specifies the amount of time in secs a route learned from a BGP peer must be stable before it is accepted into the routing database. The default value is 0, meaning that this feature is disabled. Specify a number equal to or greater than 0.

out-delay <*num*>
Used to dampen route fluctuations. Out delay is the amount of time in secs a route must be present in the routing table before it is exported to BGP. The default value is 0, meaning that this feature is disabled. Specify a number equal to or greater than 0.

keep all | none
Used to retain routes learned from a peer even if the routes' AS paths contain one of our exported AS numbers.

show-warnings
This option causes GateD to issue warning messages when receiving questionable BGP updates such as duplicate routes and/or deletions of non-existing routes. Normally these events are silently ignored.

no-aggregator-id
This option causes GateD to specify the router ID in the aggregator attribute as zero (instead of its router ID) in order to prevent different routers in an AS from creating aggregate routes with different AS paths.

keep-alives-always
This option causes GateD to always send keepalives, even when an update could have correctly substituted for one. This allows interoperability with routers that do not completely obey the protocol specifications on this point.

v3-asloop-okay
By default GateD will not advertise routes whose AS path is looped (i.e. with an AS appearing more than once in the path) to version 3 external peers. Setting this flag removes this constraint. Ignored when set on internal groups or peers.

no-v4-asloop
Prevents routes with looped AS paths from being advertised to version 4 external peers. This can be useful to avoid advertising such routes to peer which would incorrectly forward the routes on to version 3 neighbors.

as-count <*num*>
This option determines how many times we will insert our own AS number when we send the AS path to an external neighbor. Specify a number equal to or greater than 0. The default is 1. Higher values are typically used to bias upstream neighbors' route selection. (All things being equal most routers will prefer to use routes with shorter AS Paths.

Using **ascount**, the AS Path the SSR sends can be artificially lengthened.) Note that ascount supersedes the **no-v4-asloop** option--regardless of whether **no-v4-asloop** is set, the SSR will still send multiple copies its own AS if the **as-count** option is set to something greater than one.

Also, note that if the value of **ascount** is changed and GateD is reconfigured, routes will not be sent to reflect the new setting. If this is desired, it will be necessary to restart the peer session. Use only for external peer_hosts. Specify a number from 1-25.

log-up-down
Causes a message to be logged via the SYSLOG mechanism whenever a BGP peer enters or leaves the ESTABLISHED state.

ttl <*num*>
By default, BGP sets the IP TTL for local peers to ONE and the TTL for non-local peers to 255. This option is provided when attempting to communicate with improperly functioning routers that ignore packets sent with a TTL of ONE. Specify a number from 1-255.

optional-attributes-list <*num-or-string*>
Specifies the ID of the optional-attributes-list to be associated with this peer-group.

Restrictions

None.

bgp set preference

Purpose

Set BGP preference.

Format

bgp set preference <*num*>

Mode

Configure

Description

The **bgp set preference** command lets you set the BGP preference for the SSR.

Parameters

<*num*> Specifies the preference of routes learned from BGP. Specify a number from 0 - 255. The default preference is 170.

Restrictions

None.

bgp show aspaths

Purpose

Displays BGP AS path information

Format

bgp show aspaths <*aspath*> | **all [to-terminal | to-file]**

Mode

Enable

Description

The **bgp show aspaths** command displays information about a specified AS path or all AS paths. The AS path is listed along with the number of routes that use it.

Parameters

<*aspath*>	Displays information about the specified AS path.
all	Displays information about all AS paths.
to-terminal	Causes output to be displayed on the terminal. This is the default.
to-file	Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display information about all AS paths:

```
ssr# bgp show aspaths all
Hash  Ref   Path
0     5     IGP (Id 1)
2     1     (64900) 64901 64902 IGP (Id 3)
7     4     (64900) 64901 IGP (Id 2)
```

bgp show cidr-only

Purpose

Display routes in the BGP routing table with CIDR network masks

Format

bgp show cidr-only <*ip-addr-mask*> | **all** [**to-terminal** | **to-file**]

Mode

Enable

Description

The **bgp show cidr-only** command displays the same type of route information as the bgp show routes command. The difference is that the **bgp show cidr-only** command limits the display to CIDR routes only.

Parameters

<*ip-addr-mask*>	Displays information about the specified CIDR route.
all	Displays information about all CIDR routes.
to-terminal	Causes output to be displayed on the terminal.
to-file	Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display information all CIDR routes in the SSR's BGP route table:

```
ssr# bgp show cidr-only all
Proto       Route/Mask NextHop         ASPath
BGP          12.2.19/25 207.135.89.65   (64800) 64753 64752 64751 6379 3561 11277 IGP (Id 13805)
BGP        12.5.172/22 207.135.89.65   (64800) 64753 64752 64751 6379 3561 1 IGP (Id 173)
BGP        12.5.252/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 7018 6301 IGP (Id 926)
BGP          12.6.42/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 7018 11090 IGP (Id 979)
BGP        12.6.134/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 1 701 7314 10562 IGP (Id 388)
BGP        12.7.214/23 207.135.89.65   (64800) 64753 64752 64751 6379 5646 7018 4129 IGP (Id 31004)
```

bgp show community

Purpose

Displays routes that belong to a specified community.

Format

bgp show community community-id <*number*> **autonomous-system** <*number*> | **well-known-community [no-export | no-advertise | no-export-subconfed] | reserved-community** <*number*>**] [to-terminal | to-file]**

Mode

Enable

Description

The **bgp show community** command displays routes that belong to a specified community in a specified autonomous system.

Parameters

community-id <*number*>
Is the community identifier portion of a community split. This is combined with the autonomous-system value entered to create a value for the community attribute.

autonomous-system <*number*> Is an autonomous system number.

well-known-community
Is one of the well-known communities. Specify one of the following:

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

no-advertise
is a special community indicating that the routes associated with this attribute must not be advertised to other BGP peers.

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.)

reserved-community <*number*>
This option specifies one of the reserved communities that is not well-known. A reserved community is one that is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

to-terminal
Causes output to be displayed on the terminal. This is the default.

to-file
Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display routes that belong to community 160 in AS 64900:

```
ssr# bgp show community community-id 160 autonomous-system 64900
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

bgp show peer-as

Purpose

Displays information about TCP and BGP connections to an autonomous system.

Format

bgp show peer-as <*number*> **[to-terminal | to-file]**

Mode

Enable

Description

The **bgp show peer-as** command displays information about routers in a specified autonomous system that are peered with the SSR.

Parameters

peer-as <*number*> Is the AS number of a peer autonomous system.

to-terminal Causes output to be displayed on the terminal. This is the default.

to-file Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display information about TCP and BGP connections to autonomous system 64901:

```
ssr# bgp show peer-as 64901
group type External AS 64901 local 64900 flags <>
 peer 172.16.20.2 version 4 lcladdr (null) gateway (null)
  flags 0x20
  state 0x6 <Established>
  options 0x0 <>
  metric_out -1  preference 170  preference2 0
  recv buffer size 0  send buffer size 0
  messages in 10039 (updates 5  not updates 10034) 190863 octets
  messages out 10037 (updates 1  not updates 10036) 190743 octets
```

bgp show peer-group-type

Purpose

Displays status information about BGP peers by group.

Format

bgp show peer-group-type external | internal | igp | routing [to-terminal | to-file]

Mode

Enable

Description

The **bgp show peer-group-type** command displays status information about BGP peers according to their group.

Parameters

external	Displays status information about external peers.
internal	Displays status information about internal peers.
igp	Displays status information about igp peers.
routing	Displays status information about routing peers.
to-terminal	Causes output to be displayed on the terminal. This is the default.
to-file	Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display status information about external peers:

```
ssr# bgp show peer-group-type external
Group    Neighbor        V    AS MsgRcvd MsgSent State
external 172.16.20.2     4 64901   10045   10044 Established
BGP summary  1 peers in group type "external"
```

bgp show peer-host

Purpose

Displays status information about BGP peer hosts.

Format

bgp show peer-host <*ipaddr*> **received-routes | all-received-routes | advertised-routes [to-terminal | to-file]**

Mode

Enable

Description

The **bgp show peer-host** command displays information related to a specified BGP peer host. Three types of information can be displayed: routes received and accepted from a BGP peer host, all BGP routes (both accepted and rejected) from a peer host, and all routes the SSR has advertised to a peer host.

Parameters

<*ipaddr*>	Is the IP address of a BGP peer host
received-routes	Displays all valid BGP routes received and accepted from the specified peer host.
all-received-routes	Displays all BGP routes (both accepted and rejected) from the specified peer host.
advertised-routes	Displays all routes the **SSR** has advertised to the specified peer host.
to-terminal	Causes output to be displayed on the terminal. This is the default.
to-file	Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Examples

To display all valid BGP routes received and accepted from peer host 172.16.20.2:

```
ssr# bgp show peer-host 172.16.20.2 received-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 172.16.70/24       172.16.20.2                   64901 i
*> 172.16.220/24      172.16.20.2                   64901 i
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

To display all BGP routes (both accepted and rejected) from peer host 172.16.20.2:

```
ssr# bgp show peer-host 172.16.20.2 all-received-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
   172.16.20/24       172.16.20.2                   64901 i
*> 172.16.70/24       172.16.20.2                   64901 i
*> 172.16.220/24      172.16.20.2                   64901 i
*> 192.68.20/24       172.16.20.2                   64901 i
*> 192.68.222/24      172.16.20.2                   64901 64902 i
```

Displays all routes the SSR has advertised to peer host 172.16.20.2:

```
ssr# bgp show peer-host 172.16.20.2 advertised-routes
BGP table : Local router ID is 192.68.11.1
Status codes: s suppressed  d damped  h history  * valid  > best  i -
internal
Origin codes: i - IGP  e - EGP  ? - incomplete

   Network            Next Hop        Metric LocPrf Path
*> 172.16.20/24       172.16.20.1                   i
*> 192.68.11/24       192.68.11.1                   i
```

bgp show routes

Purpose

Displays entries in the BGP routing table.

Format

bgp show routes <*ip-addr-mask*> | **all [to-terminal | to-file]**

Mode

Enable

Description

The **bgp show routes** command displays the IP address/netmask, next hop, and AS path for each BGP route.

Parameters

<*ip-addr-mask*>	Displays information about the specified route.
all	Displays information about all routes.
to-terminal	Causes output to be displayed on the terminal. This is the default.
to-file	Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display the BGP routing table:

```
ssr# bgp show routes all
Proto       Route/Mask NextHop         ASPath
BGP       172.16.70/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP      172.16.220/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP       192.68.20/24 172.16.20.2     (64900) 64901 IGP (Id 2)
BGP      192.68.222/24 172.16.20.2     (64900) 64901 64902 IGP (Id 3)
```

bgp show summary

Purpose

Displays the status of all BGP connections.

Format

bgp show summary [to-terminal | to-file]

Mode

Enable

Description

The **bgp show summary** command displays the status of all BGP peers of the SSR.

Parameters

to-terminal Causes output to be displayed on the terminal. This is the default.

to-file Causes output to be saved in the file **/gatedtrc/gated.dmp**.

Restrictions

None.

Example

To display the status of all BGP connections:

```
ssr# bgp show summary
Neighbor        V    AS MsgRcvd MsgSent      Up/Down State
172.16.20.2     4 64901   10033   10031    6d23h8m1s Established
BGP summary  1 groups  1 peers
```

bgp show sync-tree

Purpose

Displays the BGP synchronization tree.

Format

bgp show sync-tree

Mode

Enable

Description

The **bgp show sync-tree** command displays the BGP synchronization tree. The synchronization tree is used by IBGP peers to resolve the next hop (forwarding address). It gives information about routes that are orphaned because the next hop could not be resolved.

Parameters

None.

Restrictions

None.

Examples

The following example shows the next hops for some of the routes that are not resolved (by showing orphaned routes):

```
ssr# bgp show sync tree
Task BGP_Sync_64805:
        IGP Protocol: Any       BGP Group: group type Routing AS 64805

        Sync Tree (* == active  + == active with alternate  - ==
inactive with alternate:
        Orphaned routes
                Forwarding address 172.23.1.18
                        3/255 peer 172.23.1.26 preference 170
                        128.36/255.255 peer 172.23.1.26 preference 170
                        128.152/255.255 peer 172.23.1.26 preference 170
                        129.200/255.255 peer 172.23.1.26 preference 170
                        129.253/255.255 peer 172.23.1.26 preference 170
                        130.44/255.255 peer 172.23.1.26 preference 170
                        130.50/255.255 peer 172.23.1.26 preference 170
                        130.132/255.255 peer 172.23.1.26 preference 170
                        134.54/255.255 peer 172.23.1.26 preference 170
                        134.120/255.255 peer 172.23.1.26 preference 170
                        134.173/255.255 peer 172.23.1.26 preference 170
                        134.217/255.255 peer 172.23.1.26 preference 170
                        134.244/255.255 peer 172.23.1.26 preference 170
                        136.1/255.255 peer 172.23.1.26 preference 170
                        137.49/255.255 peer 172.23.1.26 preference 170
                        137.159/255.255 peer 172.23.1.26 preference 170
                        138.239/255.255 peer 172.23.1.26 preference 170
```

The following example shows the next hop for all the routes that are resolved.:

```
ssr# bgp show sync-tree
Task BGP_Sync_64805:
        IGP Protocol: Any       BGP Group: group type Routing AS 64805

        Sync Tree (* == active  + == active with alternate  - ==
inactive with alternate:
        Node 3/8388608 route 3/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 4/8388608 route 4/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 6/8388608 route 6/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 9.2/32768 route 9.2/255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 9.20/16384 route 9.20/255.255.128 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.12.1/2 route 10.12.1/255.255.255.252 metric 0 interface
        Node 10.12.1.4/2 route 10.12.1.4/255.255.255.252 metric 2 next hop 172.23.1.22
        Node 10.200.12/128 route 10.200.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.203.12/128 route 10.203.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 10.204.12/128 route 10.204.12/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12/8388608 route 12/255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.2.19/64 route 12.2.19/255.255.255.128 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.2.97/128 route 12.2.97/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.3.123/128 route 12.3.123/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.4.5/128 route 12.4.5/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.4.164/128 route 12.4.164/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.164/128 route 12.5.164/255.255.255 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.172/512 route 12.5.172/255.255.252 metric -1 next hops 172.23.1.6 172.23.1.22
        Node 12.5.252/256 route 12.5.252/255.255.254 metric -1 next hops 172.23.1.6 172.23.1.22
```

bgp start|stop

Purpose

Start or stop Border Gateway Protocol (BGP).

Format

bgp start | **stop**

Mode

Configure

Description

The **bgp** start command starts BGP on the SSR.

Parameters

start Starts BGP.

stop Stops BGP.

Restrictions

None.

bgp trace

Purpose

Set BGP trace options.

Format

bgp trace [packets | open | update | keep-alive [detail | send | receive | [group <*number*> **[peer-host** <*ipaddr*>**]]] [aspath] [local-options**
all | general | state | normal | policy | task | timer | route]

Mode

Configure

Description

The **bgp trace** command lets you set BGP trace options for the SSR.

Parameters

packets	Traces all BGP packets.
open	Traces BGP OPEN packets, which are used to establish a peer relationship.
update	Traces BGP update packets, which are used to pass network reachability information.
keep-alive	Traces BGP KEEPALIVE packets, which are used to verify reachability.
detail	Shows detailed information about the specified packets.
send	Shows the specified packets sent by the router.
receive	Shows the specified packets received by the router.
local-options	Sets trace options for this protocol only. You can specify the following:
	aspath Traces aspath related events.

all	Traces all additions, changes, and deletions to the GateD routing table.
general	Activates normal and route tracing.
state	Traces state machine transitions in the protocol
normal	Traces normal protocol occurences. (Abnormal protocol occurences are always traced.)
policy	Traces the application of protocol and user-specified policies to routes being imported and exported
task	Traces system interface and processing associated with this protocol or peer
timer	Traces timer usage by this with this protocol or peer
route	Traces routing table changes for routes installed by this protocol or peer
group	Is the group ID of the group for which tracing needs to be enabled.
peer-host	peer-host ip address for which tracing needs to be enabled.The peer-host has to be qualified by the group to which it belongs

If neither the group nor peer-host is specified then tracing is enabled for all groups and peers. If the group is specified and the peer-host is not specified then the tracing is enabled for that group. If both the peer-host and group are specified than the tracing is enabled for that peer-host in the specified group

Restrictions

None.

Chapter 6

cli Commands

The **cli** commands allow you to change the behavior of the CLI in terms of command completion and command history recall.

Command Summary

Table 6 lists the **cli** commands. The sections following the table describe the command syntax.

Table 6. cli commands

cli set command completion on | off
cli set history size <*num*> **| default | maxsize**
cli set terminal rows <*num*> **columns** <*num*>
cli show history
cli show terminal
cli terminal monitor on | off

cli set command completion

Purpose

Turn on or off command completion support.

Format

cli set command completion on | off

Mode

User and Configure

Description

The **cli set command completion** command lets you enable or disable command completion support. This command works in both User and Configure mode. When executed in Configure mode, it turns on or off command completion support for the entire system. When executed in User mode, the command affects only the current login session of the user issuing that command.

Parameters

on Turn on command completion.

off Turn off command completion.

Restrictions

None.

cli set history

Purpose

Modify command history recall characteristics.

Format

cli set history size <*num*> | **default** | **maxsize**

Mode

User and Configure

Description

The **cli set history** command lets you to set the size of the command history buffer. Each command stored in this buffer can be recalled without having the user type in the same, complete command again. By setting the size of this history buffer, one tells the router how many of the most recently executed commands should be stored. When the buffer is full, the oldest command is pushed out to make space for the newest command. The **cli set history** command works in both User and Configure mode. When executed in Configure mode, it sets the history size of the entire system. When executed in User mode, the command affects only the current login session of the user issuing that command.

Parameters

size A number specifying how many of the most recently executed commands should be kept. To disable history support, specify a size of 0. The **size** option can also take the following two keywords:

default Sets the history size to the system default.

maxsize Sets the history size to the system maximum.

Restrictions

None.

Examples

To set the history buffer size to 100 commands:

```
ssr# cli set history size 100
```

cli set terminal

Purpose

Modify current session's terminal settings.

Format

cli set terminal [columns <*num*>**] [rows** <*num*>**]**

Mode

User

Description

The **cli set terminal** command lets you modify the terminal screen size of the current session. Specifying the number of rows available on your terminal causes the system to automatically pause when screen output fills the entire screen.

Parameters

columns Number of columns for your terminal. Minimum acceptable value is 20.

rows Number of rows for your terminal. The default row size is 25. To prevent output from pausing after one screen full, set the value to 0.

Restrictions

None.

Examples

To set the number of rows to 50 lines:

```
ssr# cli set terminal rows 50
```

cli show history

Purpose

Display the command history from the current CLI session.

Format

cli show history

Mode

User

Description

The **cli show history** command shows the commands you have issued during the current CLI session. A number is associated with each command. A command's number is useful for re-entering, modifying, or negating the command.

Note: You also can perform a command history recall by entering **!*** at any command prompt.

Parameters

None.

Restrictions

None.

cli show terminal

Purpose

Display information about the current terminal settings.

Format

cli show terminal

Mode

User

Description

The **cli show terminal** command shows information about the terminal settings. The terminal settings affect the display characteristics of your CLI session.

Parameters

None.

Restrictions

None.

cli terminal monitor

Purpose

Allows the current CLI session to receive or not receive console output.

Format

cli terminal monitor on | off

Mode

Enable

Description

Some system messages are normally only sent to the management console. The **cli terminal monitor** command allows the current CLI session to also receive those messages. This command is useful only if you have a current Telnet CLI session and you want the debugging output that is normally sent to the management console to also be displayed on the Telnet session.

Parameters

on Turn on receipt of console output.

off Turn off receipt of console output.

Restrictions

None.

Chapter 7

configure Command

The **configure** command places the CLI session in Configure mode. Configure mode allows you to set and change SSR parameters.

Purpose

Enter the CLI's Configure mode.

Format

configure

Mode

Enable

Description

Enters Configure mode. To exit Configure mode, use the **exit** command.

Parameters

None.

Restrictions

To enter Configure mode, you must already be in Enable mode.

Chapter 8

copy Command

The **copy** command lets you copy a file.

Purpose

Copy configuration information or files.

Format

copy active | scratchpad | tftp-server | rcp-server | startup | <*filename*> | <*url*> **to backup-CM | active | scratchpad | tftp-server | rcp-server | startup |** <*filename*> | <*url*>

Mode

Enable

Description

The **copy** command is primarily for transferring configuration information. You can copy configuration information between the SSR and external hosts using protocols such as TFTP or RCP. Within the SSR, you can copy configuration information between the SSR file system, the scratchpad (configuration database), the active (running) configuration or the Startup configuration. You also can use the **copy** command to make backup copies of a configuration file.

If the SSR has two Control Modules, you can copy the startup configuration of the primary Control Module to the secondary Control Module.

Parameters

active Specifies information from the active configuration database (the running system configuration).

scratchpad Specifies configuration changes from the scratchpad.

tftp-server Downloads or uploads a file on a TFTP server.

rcp-server Downloads or uploads a file on an RCP server.

startup Copies the Startup configuration information stored in the Control Module's NVRAM.

<*filename*> Specifies the name of a file on the SSR's local file system (NVRAM or PCMCIA card).

<*url*> Specifies a URL. You can specify one of the following types of URLs:

- **tftp** For example, **tftp://**<*hostname*>**/**<*path*>
- **rcp** For example, **rcp://**<*username*>@<*hostname*>**/**<*path*>

backup-CM Specifies that the startup configuration be copied to the secondary Control Module. You can specify the **backup-CM** parameter only as the destination and only with **startup** as the source. When **startup** is the destination, information is copied to the secondary Control Module as well.

Restrictions

The SSR does not allow some combinations of source and destination pair. Typically, you cannot have the same location for both source and destination; for example, you cannot copy from one TFTP server directly to another TFTP server or copy from scratchpad to scratchpad.

In addition, you cannot copy directly into the active configuration from anywhere except the scratchpad. All changes to the running system must come through the scratchpad.

Examples

To copy configuration information from the scratchpad to the active database, enter the following command. This command activates all the uncommitted changes, thus immediately placing the changes into effect.

```
ssr# copy scratchpad to active
```

To copy the file `config.john` to `config.debi`:.

```
ssr# copy config.john to config.debi
```

To copy the Startup configuration to a TFTP server for backup purposes, enter the following command. The CLI prompts for the TFTP server's IP address or hostname and the filename:

```
ssr# copy startup to tftp-server
```

To copy a previously saved configuration from a TFTP server to the Startup configuration, enter the following command. Note the use of an URL to specify the TFTP server and the filename.

```
ssr# copy tftp://10.1.2.3/backup/config.org to startup
```

To copy the active configuration to a remote server using RCP, enter the following command. Notice that in this example a URL specifies the RCP user name, server, and filename.

```
ssr# copy active to rcp://john@server1/config/config.dec25
```

To copy the startup configuration of the primary Control Module to the secondary Control Module:

```
ssr# copy startup to backup-CM
```

Chapter 9

diff Command

The **diff configuration** command compares the active configuration with the specified configuration file.

Format

diff configuration <*filename*> | **startup**

Mode

Configure

Description

The **diff configuration** command compares the active configuration with the specified configuration file.

Parameters

<*filename*>
Name of a configuration file.

startup
The Startup configuration file.

Restrictions

None.

Example

To compare the active configuration with the Startup configuration file:

```
ssr# diff startup
```

Chapter 10

dhcp Commands

The **dhcp** commands allow you to configure *scopes* (sets of IP address pools and network parameters) that are to be used by Dynamic Host Configuration Protocol (DHCP) clients and apply them to interfaces on the SSR.

Command Summary

Table 7 lists the **dhcp** commands. The sections following the table describe the command syntax.

Table 7. dhcp commands

dhcp <*scope*> **attach superscope** <*superscope*>
dhcp <*scope*> **define parameters** <*parameter*> <*value*>
dhcp <*scope*> **define pool** <*ip-range*>
dhcp <*scope*> **define static-ip** <*ipaddr*> **mac-address** <*macaddr*> [<*parameter*> <*value*>]
dhcp flush
dhcp global set commit-interval <*hours*>
dhcp global set lease-database <*url*>
dhcp show binding [**active** | **expired** | **static**]
dhcp show num-clients

dhcp attach superscope

Purpose

Creates a group of scopes that share a common interface.

Format

dhcp <*scope*> **attach superscope** <*superscope*>

Mode

Configure

Description

The **dhcp attach superscope** command allows you to create a "superscope," a group of scopes that share a common physical interface. For example, you can define and group together scopes for different subnets that are accessed through a single port or VLAN.

Parameters

<*scope*> The name of a scope that was previously configured with the **dhcp define** commands.

<*superscope*>The name of the group to which the specified scope is being attached.

Restrictions

None.

Example

Consider the following example where the scopes 'client1' and 'client2' exist on the same interface. To group scopes 'client1' and 'client2' into the superscope 'allclients':

```
ssr(config)# dhcp client1 attach superscope allclients
ssr(config)# dhcp client2 attach superscope allclients
```

dhcp define parameters

Purpose

Define parameters to be used by DHCP clients.

Format

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

Mode

Configure

Description

The **dhcp define parameters** command allows you to define a set of parameters that are to be used by clients when DHCP is enabled. The client uses these parameters to configure its network environment, for example, the default gateway and DNS domain name. The DHCP server on the SSR supports parameters used by Windows 95/98/NT and MacOS clients.

Parameters

<*scope*>
The name that refers to this set of client parameters.

<*parameter*> <*value*>
You can specify one or more of the following client parameters and values:

address-mask	**(Required) Specifies the address and netmask of the scope's subnet.**

Note: The **address-mask** parameter is *required* and must be defined *before* any other client parameters are specified.

broadcast	Specify the broadcast address.
bootfile	Specify the client's boot filename.
dns-domain	Specify the DNS domain name.
dns-server	Specify the IP address of the DNS server.

gateway	Specify the IP address of the default gateway.
lease-time	Specify how long, in minutes, the lease is valid. (A lease is the amount of time that an assigned IP address is valid for a client system.)
netbios-name-server	Specify the IP address of the NetBIOS name server or WINS server.
netbios-node-type	Specify the NetBIOS node type of the client.
netbios-scope	Specify the NetBIOS scope of the client.

Restrictions

None.

Examples

The following command configures a group of network parameters for the scope 'finance':

```
ssr(config)# dhcp finance define parameters address-netmask
10.33.0.0/16 dns-server 10.3.2.1 dns-domain acme.com gateway 10.33.1.1
netbios-node-type b-node lease-time 90 netbios-name-server 10.33.44.55
netbios-scope acme-finance
```

dhcp define pool

Purpose

Define a pool of IP addresses to be used by DHCP clients.

Format

dhcp <*scope*> **define pool** <*ip-range*>

Mode

Configure

Description

The **dhcp define pool** command allows you to define a pool of IP addresses that can be used by DHCP clients. An IP address pool, along with a set of parameters defined with the **dhcp define parameters** command, make up a DHCP "scope".

Parameters

<*scope*>
A name that refers to the specified pool of addresses.

<*ip-range*>
The range of IP addresses to be used by the clients. Use a hyphen (-) to designate the range. If you have more than one pool of IP addresses to specify or if the addresses are not contiguous, specify additional addresses using multiple **dhcp define pool** commands.

Restrictions

None.

Examples

To specify the addresses between 10.1.1.1 to 10.1.1.20 as the pool of IP addresses for the scope ‘clients’:

```
ssr(config)# dhcp clients define pool 10.1.1.1-10.1.1.20
```

To specify two separate pools of IP addresses for the scope ‘clients’:

```
ssr(config)# dhcp clients define pool 10.1.1.1-10.1.1.20
ssr(config)# dhcp clients define pool 10.1.1.30-10.1.1.40
```

dhcp define static-ip

Purpose

Define a static IP address for a specific MAC address.

Format

dhcp <*scope*> **define static-ip** <*ipaddr*> **mac-address** <*macaddr*> [<*parameter*> <*value*> ...]

Mode

Configure

Description

The **dhcp define static-ip** command allows you to configure a static IP address for a specific MAC address. For example, you can define a static IP address for a printer's MAC address to ensure that the printer always receives the same IP address from the DHCP server. Static IP addresses can be used for BOOTP clients as well as DHCP clients.

If you want a single MAC address to have different static IP addresses, depending upon which subnet or interface the machine is on, you can configure different scopes with different IP addresses that map to the same MAC address.

A client configured for a static IP address inherits the client parameters that are configured for the scope. If you want to configure a specific group of parameters for a static IP address, specify those parameters with the **dhcp define static-ip** command.

Parameters

<*scope*>
A name that refers to the specified static IP address.

<*ipaddr*>
The static IP address.

<*macaddr*>
The MAC address to which the specified static IP address is to be mapped.

<parameter> <value>

Specifies the client parameters and values for this static IP address. You can specify one or more of the following client parameters and values:

broadcast	Specify the broadcast address.
bootfile	Specify the client's boot filename.
dns-domain	Specify the DNS domain name.
dns-server	Specify the IP address of the DNS server.
gateway	Specify the IP address of the default gateway.
lease-time	Specify how long, in minutes, the lease is valid. (A lease is the amount of time that an assigned IP address is valid for a client system.)
netbios-name-server	Specify the IP address of the NetBIOS name server or WINS server.
netbios-node-type	Specify the NetBIOS node type of the client.
netbios-scope	Specify the NetBIOS scope of the client.

Restrictions

None.

Examples

To specify a static IP address 10.1.44.55 to the MAC address 08:00:20:12:34:56 for the scope 'servers':

```
ssr(config)# dhcp servers define static-ip 10.1.44.55 mac-address
08:00:20:12:34:56
```

To specify a static IP address 10.1.44.55 to the MAC address 08:00:20:12:34:56 for the scope 'servers' and give it a specific default gateway address:

```
ssr(config)# dhcp servers define static-ip 10.1.44.55 mac-address
08:00:20:12:34:56 gateway 10.1.1.2
```

To define two different scopes (‘public’ and ‘private’) with two different static IP addresses (10.1.44.55 and 10.2.10.23) that map to the MAC address 08:00:20:12:34:56:

```
ssr(config)# dhcp public define static-ip 10.1.44.55 mac-address
08:00:20:12:34:56
ssr(config)# dhcp private define static-ip 10.2.10.23 mac-address
08:00:20:12:34:56
```

dhcp flush

Purpose

Forces the DHCP server to update its lease database.

Format

dhcp flush

Mode

Enable

Description

The DHCP server normally updates its lease database at the intervals specified with the **dhcp global set commit-interval** command. While the DHCP server is running, you can force the server to immediately update its lease database by using the **dhcp flush** command.

Parameters

None.

Restrictions

None.

dhcp global set commit-interval

Purpose

Configure the intervals at which the DHCP server updates the lease database.

Format

dhcp global set commit-interval <*minutes*>

Mode

Configure

Description

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.

Note: Writing to flash memory can be time-consuming if there are many clients on the network.

Parameters

commit-interval <*hours*>
The interval, in hours, that the DHCP server updates the lease database. The default value is 1 hour. You can specify a value between 1-48.

Restrictions

None.

Example

To configure the DHCP server to update the lease database once every 2 hours:

```
ssr(config)# dhcp global set commit-interval 2
```

dhcp global set lease-database

Purpose

Specify a TFTP or RCP server where the lease database is backed up.

Format

dhcp global set lease-database <*url*>

Mode

Configure

Description

By default, the SSR stores the clients' lease information (the lease database) in its flash memory. You can use the **dhcp global set lease-database** command to specify a TFTP or RCP server where the lease database is to be periodically backed up.

Parameters

lease-database <*url*>
The TFTP or RCP server where the lease-database is to be backed up.

Restrictions

None.

Examples

To configure the lease database to be on a TFTP server (10.50.89.88) with the file name 'lease-db':

```
ssr(config)# dhcp global set lease-database tftp://10.50.89.88/lease-db
```

To configure the lease database to be on an RCP server (10.50.89.89) with the user name 'john' and the file name 'lease-db':

```
ssr(config)# dhcp global set lease-database
rcp://john@10.50.89.89/lease-db
```

dhcp show binding

Purpose

Display information from the lease database.

Format

dhcp show binding [active | expired | static]

Mode

Enable

Description

The **dhcp show** ommand displays information from the lease database. If you do not specify any parameters, the DHCP server displays the entire lease database.

Parameters

active
Displays currently active leases only.

expired
Displays expired leases only.

static
Displays leases with static IP address assignments only.

Restrictions

None.

Example

To display information from the lease database:

```
ssr# dhcp show binding
IP address Hardware Address Lease Expiration     Type
---------- ---------------- ----------------     ----
10.20.1.22 00:40:05:41:f1:2d 1999-05-24 17:45:06 dynamic
10.20.1.23 00:00:b4:b1:29:9c 1999-05-24 17:45:04 dynamic
10.20.1.21 00:00:b4:b0:f4:83 1999-05-24 17:45:01 dynamic
10.20.1.20 00:80:c8:e1:20:8a 1999-05-24 09:24:30 dynamic
10.30.7.9  08:00:20:11:22:33 ---                 static
10.30.7.44 08:00:20:44:55:66 ---                 static
```

dhcp show num-clients

Purpose

Display the number of allocated bindings for the DHCP server and the maximum number allowed.

Format

dhcp show num-clients

Mode

Enable

Description

This **dhcp show** ommand displays the number of allocated bindings for the DHCP server and the maximum number allowed.

Parameters

None.

Restrictions

None.

Example

To display information:

```
ssr# dhcp show num-clients
15 current clients (253 maximum)
```

Chapter 11

dvmrp Commands

The dvmrp commands let you configure and display information about Distance Vector Multicast Routing Protocol (DVMRP) interfaces.

Command Summary

Table 8 lists the dvmrp commands. The sections following the table describe the command syntax.

Table 8. dvmrp commands

dvmrp accept | **noaccept route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*> **[router** <*IPaddr*>**]]**
dvmrp advertise | **noadvertise route** <*IPaddr/mask*> **[exact] [interface** <*IPaddr*>**]**
dvmrp create tunnel <*name*> **local** <*IPaddr*> **remote** <*IPaddr*>
dvmrp enable no-pruning
dvmrp enable interface <*IPaddr*> | <*interface-name*> | <*tunnel-name*>
dvmrp set interface <*IPaddr*> | <*hostname*> **[metric** <*num*>**] [neighbor-timeout** <*seconds*>**] [prunetime** <*seconds*>**] [rate** <*num*>**] [scope** <*IPaddr/mask*>**] [threshold** <*num*>**]**
dvmrp show interface [<*IPaddr*>**]**
dvmrp show routes host <*IPaddr*> | **interface** <*IPaddr*> | **net** <*netaddr*> | **router** <*IPaddr*>
dvmrp show rules
dvmrp start

dvmrp accept route

Purpose

Specifies routes to be accepted from DVMRP neighbor routers.

Format

dvmrp accept | noaccept route <*IPaddr/mask*> **[exact] [interface** <*IPaddr*> **[router** <*IPaddr*>**]]**

Mode

Configure

Description

The **dvmrp accept route** command allows you to specify particular routes that can be learned from DVMRP neighbors.

A route is always accepted from a DVMRP neighbor unless you use the **dvmrp noaccept route** to prevent it from being accepted. You can use the **dvmrp accept route** command along with the **dvmrp noaccept route** command to filter the routes accepted from DVMRP neighbor routers.

Parameters

accept
Allows the specified route to be accepted from DVMRP neighbor routers.

noaccept
Prevents the specified route from being accepted from DVMRP neighbor routers.

route <*IPaddr/mask*>
Is the IP address and mask of the route prefix to be accepted.

exact
Causes only routes exactly matching the prefix to be accepted.

interface <*ipAddr*>
Is the IP address of the interface to which you are applying this filter.

router <*IPaddr*>
Is the IP address of a DVMRP neighbor router.

Restrictions

None.

Examples

To cause the SSR to accept only prefix 20.30.40.0/24, and filter out all other routes:

```
ssr(config)# dvmrp noaccept route 0/0 interface customer1
ssr(config)# dvmrp accept route 20.30.40.0/24 interface customer1
```

If interface customer1 breaks subnet 20.30.40.0/24 into smaller subnets, you can filter out routes from these subnets with the following commands:

```
ssr(config)# dvmrp noaccept route 0/0 interface customer1
ssr(config)# dvmrp accept route 20.30.40.0/24 interface customer1 exact
```

dvmrp advertise route

Purpose

Specifies routes to be advertised to DVMRP neighbor routers.

Format

dvmrp advertise | noadvertise route <*IPaddr/mask*> **[exact] [interface** <*IPaddr*>**]**

Mode

Configure

Description

The **dvmrp advertise route** command allows you to specify particular routes that can be advertised to DVMRP neighbors. A route is always advertised to a DVMRP neighbor unless you use the **dvmrp noadvertise route** command to prevent it from being advertised. You can use the **dvmrp advertise route** command along with **dvmrp noadvertise route** to filter the routes advertised to DVMRP neighbor routers.

Parameters

advertise
Allows the specified route to be advertised to DVMRP neighbor routers.

noadvertise
Prevents the specified route from being advertised to DVMRP neighbor routers.

route <*IPaddr/mask*>
Is the IP address and mask of the route prefix to be advertised.

exact
Causes only routes exactly matching the prefix to be advertised.

interface <*ipAddr*>
Is the IP address of the interface to which you are applying this filter.

Restrictions

None.

Examples

To prevent route 10.0.0.0/8 from being advertised on interface mbone (all other routes are advertised):

```
ssr(config)# dvmrp noadvertise route 10/8 interface mbone
```

To advertise only route 20.20.20.0/24 to its neighbors on interface mbone:

```
ssr(config)# dvmrp noadvertise route 0/0 interface mbone
ssr(config)# dvmrp advertise route 20.20.20.0/24 interface mbone
```

dvmrp create tunnel

Purpose

Creates a DVMRP tunnel.

Format

dvmrp create tunnel <*name*> **local** <*ipAddr*> **remote** <*ipAddr*>

Mode

Configure

Description

The **dvmrp create tunnel** command creates a DVMRP tunnel for sending multicast traffic between two end points.

Parameters

<*name*>	Name of this DVMRP tunnel.
local <*ipAddr*>	IP address of the local end point of this tunnel. **Note:** The local IP address must already be configured on the SSR.
remote <*ipAddr*>	IP address of the remote end point of this tunnel.

Restrictions

- Tunnels use unicast routing principles. Make sure a route exists between the tunnel source and destination (***local*** <*ipAddr*> and ***remote*** <*ipAddr*>) you specify.
- An IP interface has to exist before a tunnel can be created from it.

  Note: A good way to confirm that a tunnel exists is to ping the other end of the tunnel.
- Tunnels cannot be created between two endpoints (that is, on the same subnet).
- A maximum of eight tunnels are allowed.

-

Example

To create a DVMRP tunnel called *tun12* between 10.3.4.15 (the local end of the tunnel) and 10.5.3.78 (the remote end of the tunnel):

```
ssr(config)# dvmrp create tunnel tun12 local 10.3.4.15 remote 10.5.3.78
```

dvmrp enable no-pruning

Purpose

Disables DVMRP pruning.

Note: Pruning is enabled by default. The current DVMRP specification requires pruning capability. Unless you have a good reason for disabling pruning, Cabletron Systems recommends that you leave it enabled.

Format

dvmrp enable no-pruning

Mode

Configure

Description

Disable DVMRP pruning.

Parameters

None.

Restrictions

None.

dvmrp enable interface

Purpose

Enables DVMRP on an interface.

Format

dvmrp enable interface <*ipAddr/name*> | <*tunnel-name*>

Mode

Configure

Description

The **dvmrp enable interface** command enables DVMRP on the specified interface.

Parameters

<*ipAddr/name*> | <*tunnel-name*>
IP address or tunnel name of the interface on which you are enabling DVMRP.

- If you are enabling DVMRP on an interface that does not have a tunnel, specify its name or IP address.
- If you are enabling DVMRP on an interface that has a tunnel, specify the tunnel name.

Restrictions

Note: The Control Module's en0 interface is never used for multicast traffic.

DVMRP does not run on multiple IP subnets if created on an interface. Currently, the SSR automatically picks up the first subnet to run DVMRP on it. However any one particular subnet can be picked up by enabling it. But before doing that, no subnet should already be enabled on that interface. The SSR supports a maximum of 64 DVMRP and IGMP interfaces.

Note: The **igmp enable interface** command has a similar restriction of using only one subnet.

Examples

To enable DVMRP on the IP interface with IP address 10.50.78.2:

```
ssr(config)# dvmrp enable interface 10.50.78.2
```

To enable tunnel tun12:

```
ssr(config)# dvmrp enable interface tun12
```

dvmrp set interface

Purpose

Configures various DVMRP parameters on an interface.

Format

dvmrp set interface <*IPaddr/name*> **[metric** <*num*>**] [neighbor-timeout** <*seconds*>**] [prunetime** <*seconds*>**] [rate** <*num*>**] [scope** <*IPaddr/mask*>**] [threshold** <*num*>**]**

Mode

Configure

Description

The **dvmrp set interface** command sets DVMRP parameters on an IP interface.

Parameters

<*ipAddr/name.*
IP address or name of the interface on which you are configuring DVMRP parameters.

metric <*num*>
The metric (cost) of this interface. Specify a number in the range 1 – 16. The default is 1. Normally you should not change this setting unless the network topology requires it.

neighbor-timeout <*num*>
The number of seconds after which the SSR will consider the neighbor to be down. Specify a number in the range 40 – 400. The default is 35.

Note: If you have some old routers, this value should be increased to accommodate them because they don't send probes or route updates at 40-second intervals.

prunetime <*seconds*>
The multicast prunetime of this interface. Specify a number in the range 300 – 7200. The default is 3600 seconds (one hour).

rate <*num*>
The multicast rate of this interface in kbps. Specify a number in the range 1 – 10000. The default is 500.

Note: The option applies only to tunnels.

scope <*IPaddr/mask*>
The multicast scope of this interface. The purpose of this option is to disallow the groups specified by a scope from being forwarded across an interface. This option therefore is a filtering mechanism. The threshold and the scope are two common mechanisms for implementing local simple filtering of a multicasting data.

Specify an IP address and network mask. Examples: 230.2.3.4/255.255.0.0 or 230.2.3.4/16.

threshold <*num*>
The multicast threshold of this interface. The purpose of this option is to allow forwarding of a packet on a multicast interface only if the packet's threshold is at least the configured value. The threshold and the scope are two common mechanisms for implementing local simple filtering of a multicasting data.

Specify a number in the range 1 – 255. The default is 1.

Restrictions

None.

Examples

To configure the interface 10.50.89.90 to have a metric of 5 and a threshold of 16:

```
ssr(config)# dvmrp set interface 10.50.89.90 metric 5 threshold 16
```

dvmrp show interface

Purpose

Displays DVMRP interfaces.

Format

dvmrp show interface [<*IPaddr*>]

Mode

Enable

Description

The **dvmrp show interface** command displays the state of an interface running DVMRP, along with other neighbor-related information. Neighbors are displayed with their DVMRP version and capability flags and Generation IDs; this information can help in debugging. If rules are in effect for an interface, they are indicated by `ExportPol` or the `ImportPol` flags.

Parameters

<*IPaddr*> Displays DVMRP information for the specified interface.

Restrictions

None.

Examples

Here is an example of the **dvmrp show interface** command.

```
ssr# dvmrp show interface
Address: 10.50.1.1            Subnet: 10.50.1/24        Met: 1   Thr: 1
Name   : pc                    State: Dn   Igmp  Dvmrp

Address: 207.135.89.10        Subnet: 207.135.89.0/27   Met: 1   Thr: 1
Name   : corp                  State: Up   Igmp  Dvmrp Querier  ExportPol
Peer   : 207.135.89.1        Version:   3.255         Flags:0xe  GID: 0x31a

Address: 10.55.89.101         Subnet: 10.55.89/24       Met: 1   Thr: 1
Name   : lab                   State: Up   Dvmrp
Peer   : 10.55.89.100        Version:   3.255         Flags:0xe  GID: 0x179

Address: 207.135.89.10       Remote: 207.137.137.1     Met: 1   Thr: 1  Rate: 1000
Name   : mbone                 State: Tunnel  Up   Dvmrp ExportPol
Peer   : 207.137.137.1       Version:   3.8           Flags:0xe  GID: 0x6c19d135
```

dvmrp show routes

Purpose

Displays DVMRP unicast routing table.

Format

dvmrp show routes host <*IPaddr*> | **interface** <*IPaddr*> | **net** <*netaddr*> | **router** <*IPaddr*> **subordinates** | **permission**

Mode

Enable

Description

The **dvmrp show routes** command displays the contents of DVMRP unicast routing table.

DVMRP routes show the topology information for the internet multicasting sites. It is independent of IP unicast routing table or protocol. In this table, the information is presented about a address prefix (in form of network-address/network-mask length), the interface and the uplink (parent) router through which this subnet can be reached. This table also shows information about any routers/interfaces which consider this router as their uplink (that is, those routers which depend on this router if traffic were to originate from this subnet). These routers/interfaces are shown as children of the parent router.

Note: The **dvmrp show routes** command can search on the basis of subnet and on the basis of those routes whose parent is a particular interface and/or a particular router.

Note: This command only shows DVMRP routes and not information about current multicast sessions. For information about current multicast sessions, use the **multicast show mroutes** command.

Parameters

host <*IPaddr*> Displays the route to the specified uplink host address.

interface <*IPaddr*> Displays the interface address of the specified uplink interface.

net <*netaddr*>	Displays the route to the specified prefix (or subnets falling within the prefix).
router <*IPaddr*>	Displays the route to the specified router.
subordinates	Displays the downstream routers list.
permissions	Indicates whether a route is affected by any rules. Routes marked NoAdv are not advertised.

Restrictions

None.

Examples

To display DVMRP routes offered by the next-hop router 207.137.137.1:

```
ssr# dvmrp show routes router 207.137.137.1
DVMRP Routing Table (4232 routes    8 hold-down-routes)
Net: 128.119.3.16/29         Gateway: 207.137.137.1     Met:  9   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 128.119.3.8/29          Gateway: 207.137.137.1     Met:  9   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 209.12.162.16/28        Gateway: 207.137.137.1     Met: 26   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.197.171.112/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.240/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.192/28      Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
                                      lab

Net: 208.151.215.96/28       Gateway: 207.137.137.1     Met:  7   Age:  35
Parent: mbone               Children: corp
```

To show non-advertised routes on interface lab:

```
ssr# dvmrp show routes interface lab permission
DVMRP Routing Table (4232 routes    5 hold-down-routes)
Net: 100.100.100/24          Gateway: 10.55.89.100      Met:  2   Age:  25
Parent: lab                 Children: corp
                                      mbone             leaf NoAdv

Net: 20.20.20/24             Gateway: 10.55.89.100      Met:  2   Age:  25
Parent: lab                 Children: corp
                                      mbone             leaf NoAdv

Net: 10.55.89/24             Gateway: ----              Met:  1   Age:  --
Parent: lab                 Children: corp              leaf NoAdv
                                      mbone             leaf NoAdv

Total Routes Printed: 3
```

dvmrp show rules

Purpose

Displays the rules in effect for filtering routes from DVMRP neighbor routers.

Format

dvmrp show rules

Mode

Enable

Description

The **dvmrp show rules** command displays the filtering rules in effect for DVMRP routes. Once you have set rules with the **dvmrp accept** and **dvmrp advertise** commands, you can display the active rules by entering the **dvmrp show rules** command.

Parameters

None.

Restrictions

None.

Example

In this example, the following rules are in effect:

```
dvmrp advertise route 207.135.89.0/24 interface mbone
dvmrp noadvertise route 0/0 interface mbone
dvmrp advertise route 207.135.88.0/24 interface mbone
dvmrp noadvertise route 10/8 interface corp
```

To display information about these rules:

```
# dvmrp show rules
NoAdvertise: 10.0.0.0/8                IF: corp
Advertise  : 207.135.89.0/24           IF: mbone
Advertise  : 207.135.88.0/24           IF: mbone
NoAdvertise: default                   IF: mbone
```

These rules would affect the routing table as follows:

```
# dvmrp show route net 10/8 permissions

Net: 10.55.89/24           Gateway: ----            Met:  1   Age:  --
Parent: lab                Children: corp             leaf NoAdv
                                     mbone            leaf NoAdv
```

These rules prevent a directly connected route on this router from being visible to interface corp and mbone. The `leaf` flag indicates there is no downstream neighbor on the interface.

dvmrp start

Purpose

Starts DVMRP multicast routing.

Format

dvmrp start

Mode

Configure

Description

The **dvmrp start** command starts DVMRP multicast routing on the configured multicast-enabled interfaces and tunnels.

Note: Because DVMRP is the only multicasting protocol on the SSR, IGMP starts and stops along with DVMRP. If you want to start IGMP on local interfaces, you still must use this command.

DVMRP is by default not running. DVMRP does not interact with any unicast protocol. However if you need to run a tunnel, make sure that the tunnel is reachable by a unicast routing mechanism.

Parameters

None.

Restrictions

None.

Chapter 12

enable Command

The **enable** command switches the CLI session from User mode to Enable mode.

Format

enable

Mode

User

Description

The **enable** command switches your CLI session from User mode to Enable mode. After you issue the command, the CLI will prompt you for a password if a password is configured. If no password is configured, a warning message advising you to configure a password is displayed.

If a password is configured and you do not know your password or pressing Return does not work, see the administrator for the SSR.

To exit from the Enable mode and return to the User mode, use the **exit** command. To proceed form the Enable mode into the Configure mode, use the **configure** command.

Parameters

None.

Restrictions

None.

Chapter 13

erase Command

The **erase** command erases the contents of the scratchpad or Startup configuration files.

Format

erase scratchpad | startup

Mode

Configure

Description

The **erase scratchpad** command erases the contents of the SSR's command scratchpad. The **erase startup** command erases the Startup configuration from the Control Module's NVRAM.

Parameters

scratchpad	Erases the contents of the scratchpad. The scratchpad contains configuration commands that you have issued but have not yet activated.
startup	Erases the contents of the Startup configuration. The Startup configuration is the configuration the SSR uses to configure itself when you reboot it. When you erase the Startup configuration, then reboot immediately, the SSR restarts without any configuration information.

Restrictions

The erase commands do not delete other types of files. To delete a file, use the **file del** command.

Chapter 14

exit Command

The **exit** command exits the current CLI mode to the previous mode. For example, if you are in the Enable mode, **exit** returns you to the User mode. If you are in Configure mode, **exit** returns you to Enable mode. If you are in User mode, **exit** closes your CLI session and logs you off the SSR.

Format

exit

Mode

All modes.

Parameters

None.

Restrictions

None.

Chapter 15

file Commands

The **file** commands enable you to display a directory of the files on a storage device, display the contents of a file on the console, and delete a file.

Command Summary

Table 9 lists the **file** commands. The sections following the table describe the command syntax.

Table 9. file commands

file delete <*file-name*>
file dir <*device-name*>
file type <*file-name*>

file delete

Purpose

Delete a file.

Format

file delete <*file-name*>

Mode

Enable

Description

The **file delete** command deletes the specified file. The filename can include a device name. By default, if a device name is not specified, it is assumed to be the **bootflash:** device which is where all configuration files are stored.

Parameters

<*file-name*>	Name of the file to delete. The filename can include a device name using this format: <*device*>**:**<*file-name*>. By default, if a device name is not specified, it is assumed to be the **bootflash** device. The **bootflash** device is the default device for storing configuration files.

Restrictions

None.

Examples

To delete the file `config.old`:

```
ssr# file delete config.old
```

file dir

Purpose

Display contents of a file system.

Format

file dir <*device-name*>

Mode

User.

Description

Displays a directory of the files on the specified storage device.

Parameters

<*device-name*> Device name. You can specify one of the following:

bootflash: The Control Module's NVRAM.

slot0: The PCMCIA flash card in slot 0 (the upper slot).

slot1: The PCMCIA flash card in slot 1(the lower slot).

Restrictions

None.

Examples

To display the contents of the **bootflash** device:

```
ssr# file dir bootflash:
```

file type

Purpose

Display contents of a file.

Format

file type <*file-name*>

Mode

Enable.

Description

Displays the contents of a file.

Parameters

<*file-name*> Name of the file to display. The filename can include a device name using this format: <*device*>**:**<*file-name*>. By default, if a device name is not specified, it is assumed to be the **bootflash** device. The **bootflash** device is the default device for storing configuration files.

Restrictions

None.

Examples

To display the contents of the file `startup` (the startup configuration file):

```
ssr# file type startup
```

Chapter 16

filters Commands

The **filters** commands let you create and apply the following types of security filters:

- **Address filters.** Address filters block traffic based on a frame's source MAC address, destination MAC address, or both. Address filters are always configured and applied on the input port.
- **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. Static entry filters are always configured and applied on the input port. You can configure source static entry filters, destination static entry filters, and flow static entry filters. Source static entry filters allow or disallow frames based on their source MAC address; destination static entry filters allow or disallow frames based on their destination MAC address. Flow static entries allow or disallow traffic based on their source *and* destination MAC addresses.
- **Port-to-address locks.** Port-to-address lock filters "lock" a user to a port or set of ports, disallowing them access to other ports.
- **Secure ports.** Secure port filters shut down Layer 2 access to the SSR from a specific port or drop all Layer 2 packets received by a port. Used by themselves, secure ports secure unused SSR ports. When used in conjunction with static entry filters, secure ports drop all received or sent traffic (depending on the static entry filter) except traffic forced to or from the port by the static entry filter.

Command Summary

Table 10 lists the filters commands. The sections following the table describe the command syntax.

Table 10. filters commands

filters add address-filter name <*name*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>
filters add port-address-lock name <*name*> **source-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>
filters add secure-port name <*name*> **direction source | destination vlan** <*VLAN-num*> **in-port-list** <*port-list*>
filters add static-entry name <*name*> **restriction allow | disallow | force source-mac** <*MACaddr*> **dest-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*> **out-port-list** <*port-list*>
filters show address-filter [all-source | all-destination | all-flow] **[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**] [ports** <*port-list*>**]** **[vlan** <*VLAN-num*>**]**
filters show port-address-lock ports [ports <*port-list*>**]** **[vlan** <*VLAN-num*>**] [source-mac** <*MACaddr*>**]**
filters show secure-port
filters show static-entry [all-source | all-destination | all-flow] **ports** <*port-list*> **vlan** <*VLAN-num*> **[source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>**]**

filters add address-filter

Purpose

Applies an address filter.

Format

filters add address-filter name <*name*> **source-mac** <*MACaddr*>
dest-mac <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>

Mode

Configure

Description

The **filters add address-filter** command blocks traffic based on a frame's source MAC address (**source-mac**), destination MAC address (**dest-mac**), or a flow (specified using both a source MAC address and a destination MAC address).

Parameters

name <*name*>	Specifies the name of the filter.
source-mac <*MACaddr*>	Specifies the source MAC address. Use this option for source or flow address filters.
dest-mac <*MACaddr*>	Specifies the destination MAC address. Use this option for destination or flow static entries.
vlan <*VLAN-num*>	Specifies the VLAN.
in-port-list <*port-list*>	Specifies the ports to which you want to apply the filter.

Restrictions

You should apply flow filters (specified using both a source MAC address and a destination MAC address) only to ports that are using flow-based bridging.

filters add port-address-lock

Purpose

Applies a port address lock.

Format

filters add port-address-lock name <*name*> **source-mac** <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>

Mode

Configure

Description

The **filters add port-address-lock** command locks a user (identified by the user's MAC address) to a specific port or set of ports. The source MAC address will be allowed to reach only those stations and other ports that are connected to a port specified by **in-port-list**.

Parameters

name <*name*>	Specifies the name of the lock filter.
source-mac <*MACaddr*>	Specifies the source MAC address.
vlan <*VLAN-num*>	Specifies the VLAN.
in-port-list <*port-list*>	Specifies the ports to which you want to apply the lock.

Restrictions

None.

filters add secure-port

Purpose

Applies a port security filter.

Format

filters add secure-port name <*name*> **direction source | destination vlan** <*VLAN-num*> **in-port-list** <*port-list*>

Mode

Configure

Description

The **filters add secure-port** command shuts down Layer 2 access to the SSR from the ports specified by **in-port-list**. The SSR drops all traffic received from these ports.

Note: You can use port-to-address lock filters to force traffic to a port secured by the **filters add secure-port** command.

Parameters

name <*name*>
Specifies the name of the filter.

direction source | destination
Specifies whether the filter is to secure a source port or a destination port.

vlan <*VLAN-num*>
Specifies the VLAN.

in-port-list <*port-list*>
Specifies the ports to which you want to apply the filter.

Restrictions

None.

filters add static-entry

Purpose

Applies a static entry.

Format

filters add static-entry name <*name*>
restriction allow | disallow | force source-mac <*MACaddr*>
dest-mac <*MACaddr*> **vlan** <*VLAN-num*> **in-port-list** <*port-list*>
out-port-list <*port-list*>

Mode

Configure

Description

The **filters add static-entry** command allows, disallows, or forces traffic to go to a set of destination ports based on a frame's source MAC address (**source-mac**), destination MAC address (**dest-mac**), or a flow (specified using both a source MAC address and a destination MAC address).

Parameters

name <*name*>
Specifies the name of the static-entry filter.

restriction allow | disallow | force
Specifies the forwarding behavior of the static entry, which can be one of the following keywords:

allow Allows packets to go to the set of ports specified by out-port-list.

disallow Prohibits packets from going to the set of ports specified by out-port-list.

force Forces packets to go to the set of ports specified by out-port-list, despite any port locks in effect on the ports.

source-mac <*MACaddr*>
Specifies the source MAC address. Use this option for source or flow static entries.

dest-mac <*MACaddr*>
Specifies the destination MAC address. Use this option for destination or flow static entries.

in-port-list <*port-list*>
Specifies the ports to which you want to apply the static entry.

out-port-list <*port-list*>
Specifies the ports to which you are allowing, disallowing, or forcing packets.

Restrictions

You should apply flow filters (specified using both a source MAC address and a destination MAC address) only to ports that are using flow-based bridging.

filters show address-filter

Purpose

Displays the address filters.

Format

filters show address-filter
[all-source | all-destination | all-flow]
[source-mac <*MACaddr*> **dest-mac** <*MACaddr*>**] [ports** <*port-list*>**]**
[vlan <*VLAN-num*>**]**

Mode

Enable

Description

The **filters show address-filter** command displays the address filters currently configured on the SSR.

Parameters

all-source | all-destination | all-flow
Specifies the types of filters you want to display.

source-mac <*MACaddr*>
Restricts the display to only those address filters that have been applied to this source MAC address.

dest-mac <*MACaddr*>
Restricts the display to only those address filters that have been applied to this destination MAC address.

ports <*port-list*>
Restricts the display to only those address filters that have been applied to the specified ports.

vlan <*VLAN-num*>
Restricts the display to only those address filters that have been applied to the specified VLANs.

Restrictions

None.

filters show port-address-lock

Purpose

Display the port address locks.

Format

filters show port-address-lock [ports <*port-list*>**]**
[vlan <*VLAN-num*>**] [source-mac** <*MACaddr*>**]**

Mode

Enable

Description

The **filters show port-address-lock** command displays the port-address-lock filters currently configured on the SSR.

Parameters

ports <*port-list*>
Restricts the display to only those port address locks that have been applied to the specified ports.

vlan <*VLAN-num*>
Restricts the display to only those port address locks that have been applied to the specified VLANs.

source-mac <*MACaddr*>
Restricts the display to only those port address locks that have been applied to this source MAC address.

Restrictions

None.

filters show secure-port

Purpose

Display the port security filters.

Format

filters show secure-port

Mode

Enable

Description

The **filters show secure-port** command displays the secure-port filters currently configured on the SSR.

Parameters

None.

Restrictions

None.

filters show static-entry

Purpose

Displays the static entry filters.

Format

filters show static-entry [all-source | all-destination | all-flow]
ports <*port-list*> **vlan** <*VLAN-num*>
[source-mac <*MACaddr*> **dest-mac** <*MACaddr*>**]**

Mode

Configure

Description

The **filters show static-entry** command displays the static-entry filters currently configured on the SSR.

Parameters

all-source | all-destination | all-flow
Specifies the types of static entries you want to display.

ports <*port-list*>
Restricts the display to only those static entries that have been applied to the specified ports.

vlan <*VLAN-num*>
Restricts the display to only those static entries that have been applied to the specified VLANs.

source-mac <*MACaddr*>
Restricts the display to only those static entries that have been applied to this source MAC address.

dest-mac <*MACaddr*>
Restricts the display to only those static entries that have been applied to this destination MAC address.

Restrictions

None.

Chapter 17

frame relay Commands

The following commands allow you to define frame relay service profiles, and specify and monitor frame relay High-Speed Serial Interface (HSSI) and standard serial ports.

Command Summary

Table 11 lists the frame relay commands. The sections following the table describe the command syntax.

Table 11. frame relay commands

frame-relay apply service <*service name*> **ports** <*port list*>
frame-relay create vc <*port*>
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]**
frame-relay set fr-encaps-bgd ports <*port list*>
frame-relay set lmi [error-threshold <*number*>**] [full-enquiry-interval** <*number*>**] [monitored-events** <*number*>**] [polling-interval** <*number*>**] [state enable | disable] [type ansi617d-1994 | q933a | rev1] port** <*port list*>
frame-relay set payload-compression [type frf9_mode1_stac] port <*port list*>
frame-relay set peer-addr <*IP address*> **ports** <*port list*>

Table 11. frame relay commands (Continued)

frame-relay show service <*service name*> | **all**
frame-relay show stats port <*port name*> **[last-error] [lmi] [mibII]**
frame-relay show stats port <*port name*> **summary**

frame-relay apply service ports

Purpose

Apply a pre-defined service profile to a frame relay virtual circuit (VC).

Format

frame-relay apply service <*service name*> **ports** <*port list*>

Mode

Configure

Description

Issuing the **frame-relay apply service** command allows you to apply a previously defined service profile to a given frame relay VC.

Parameters

<*service name*>	The name of the previously defined service profile you wish to apply to the given port(s) or interfaces.
<*port list*>	The port(s) to which you wish to apply the pre-defined service profile. You can specify a single VC or a comma-separated list of VCs.

Restrictions

Usage is restricted to frame relay VCs only.

Example

To apply the service "s1" to slot 2, VC 100 on serial ports 1 and 2:

```
ssr(config)# frame-relay apply service s1 ports se.2.1.100 se.2.2.100
```

frame-relay create vc

Purpose

Create frame relay virtual circuits (VCs).

Format

frame-relay create vc <*port*>

Mode

Configure

Description

The **frame-relay create vc** command allows you to create a frame-relay virtual circuit on a slot and port location specified in the command line.

Parameters

<*port*> The port on which you wish to create a frame relay virtual circuit.

Restrictions

Usage is restricted to frame relay ports only.

Example

To create a frame relay virtual circuit with a DLCI of 100 on serial port 1 of slot 3:

```
ssr(config)# frame-relay create vc port se.3.1.100
```

frame-relay define service

Purpose

Configure service profiles for frame relay ports.

Format

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]**

Mode

Configure

Description

The **frame-relay define service** command allows you to specify the following attributes for a newly created service profile:

- Number of bits per second contained in a committed burst for frame relay virtual circuits.
- Number of bits per second contained in an excessive burst for frame relay virtual circuits.
- Whether or not to simultaneously enable and specify the threshold at which adaptive shaping will activate when receiving BECN frames
- The committed information rate (in bits per second) for frame relay virtual circuits.
- The allowable queue depth for high-, low-, and medium-priority frames on frame relay VCs.
- Activation or deactivation of Random Early Discard (RED) for frame relay circuits.

- The maximum and minimum threshold values for RED high-, low-, and medium-priority traffic.

  In general, Cabletron recommends that the maximum threshold values be less than or equal to the respective high-, low-, or medium-priority queue depth. The minimum threshold values should be one-third of the respective maximum threshold.

- Activation and deactivation of RMON for frame relay VCs. Note that before you can view RMON statistics such as Ethernet statistics and history for frame relay ports, RMON has to be activated.

Parameters

<*service name*>
The name you wish to assign to the newly created service profile.

Bc <*number*>
The number of bits per second contained in a committed burst for a frame relay virtual circuit. You can specify a number between 1 and 2,147,483,646 bits per second.

Be <*number*>
The number of bits per second contained in an excessive burst for a frame relay virtual circuit. You can specify a number between 1 and 2,147,483,646 bits per second.

becn-adaptive-shaping <*number*>
The threshold (number of frames) at which adaptive shaping will activate when receiving BECN frames. You can specify a number between 1 and 100,000 frames.

cir <*number*>
The committed information rate (in bits per second) for frame relay virtual circuits. You can specify a number between 1 and 2,147,483,646 bits.

high-priority-queue-depth <*number*>
The number of high-priority frames allowed in the frame relay queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

low-priority-queue-depth <*number*>
The number of low-priority frames allowed in the frame relay queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

med-priority-queue-depth <*number*>
The number of medium-priority frames allowed in the frame relay queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

red on | off
Specifying the **on** keyword enables RED for frame relay ports. Specifying the **off** keyword disables RED for frame relay ports.

red-maxTh-high-prio-traffic <*number*>
The maximum allowable number of frames for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-maxTh-low-prio-traffic <*number*>
The maximum allowable number of frames for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-maxTh-med-prio-traffic <*number*>
The maximum allowable number of frames for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-minTh-high-prio-traffic <*number*>
The minimum allowable number of frames for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

red-minTh-low-prio-traffic <*number*>
The minimum allowable number of frames for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

red-minTh-med-prio-traffic <*number*>
The minimum allowable number of frames for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

rmon on | off
Specifying the **on** keyword enables RMON for frame relay VCs. Specifying the **off** keyword disables RMON for frame relay VCs.

Restrictions

When defining a value for **bc**, you *must* also be sure to define an appropriate value for **cir**, and vice-versa.

Examples

Suppose you wish to specify a frame relay virtual circuit with the following attributes:

- Committed burst value of 35 million and excessive burst value of 30 million
- BECN active shaping at 65 thousand frames
- Committed information rate (CIR) of 120 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:

```
ssr(config)# frame-relay define service profile1 Bc 35000000 Be 30000000
   becn-adaptive-shaping 65000 cir 120000000 red off rmon on
```

frame-relay set fr-encaps-bgd

Purpose

Force the ingress packets to be encapsulated in bridged format.

Format

frame-relay set fr-encaps-bgd ports <*port list*>

Mode

Configure

Description

Issuing the **frame-relay set fr-encaps-bgd** command allows you to use bridged format encapsulation on a given frame relay VC.

Parameters

<*port list*> The port(s) to which you wish to use bridged encapsulation. You can specify a single VC or a comma-separated list of VCs.

Restrictions

Usage is restricted to frame relay VCs only.

Example

To force the bridged encapsulation to slot 2, VC 100 on serial ports 1 and 2:

```
ssr(config)# frame-relay fr-encaps-bgd ports se.2.1.100 se.2.2.100
```

frame-relay set lmi

Purpose

Set frame relay Local Management Interface (LMI) parameters.

Format

frame-relay set lmi [error-threshold <*number*>**] [full-enquiry-interval** <*number*>**] [monitored-events** <*number*>**] [polling-interval** <*number*>**] [state enabled | disabled] [type ansi617d-1994 | q933a | rev1] port** <*port list*>

Mode

Configure

Description

The **frame-relay set lmi** command allows you to specify the following attributes:

- The number of times the router will attempt to poll an LMI interface before declaring it down. You can define a value between 1 and 10, inclusive.
- The number of status enquiries that will be sent before a full status enquiry is requested. You can define a value between 1 and 255, inclusive.
- The number of status enquiries over which various pieces of LMI information can be collected and tabulated. For example, you can tabulate the number of times an interface was declared down/lost due to a lack of proper responses to status enquiries. You can define a value between 1 and 10, inclusive.
- The number of seconds that pass between successive status enquiry messages. You can define a value between 5 and 30, inclusive.
- Whether or not LMI messages are sent. LMI messages are not sent by default.
- The LMI type for frame relay WAN ports.

Parameters

error-threshold <*number*>
The number of unanswered status enquiries that the router will make before declaring an interface to be down.

full-enquiry-interval <*number*>
The number of status enquiries that will be sent before a full report on status is compiled and transmitted.

monitored-events <*number*>
The number of status enquiries over which collection and tabulation of various pieces of LMI information will take place.

polling-interval <*number*>
The amount of time (in seconds) that will pass before a subsequent status enquiry takes place.

state enabled | disabled
Enables the sending and receiving of LMI messages. If LMI messages are enabled, the operational status of each VC is determined by the LMI messages. If LMI messages are disabled, each VC is assumed to be operationally "up". LMI messages are disabled by default.

type ansi617d-1994 | q933a | rev1
The LMI type for frame relay WAN ports. You can only specify the **ansi617d-1994**, **q933a**, or **rev1** keywords to define as the LMI type for WAN ports.

port <*port list*>
The port or ports that will assume the LMI service profile behavior.

Restrictions

None.

Examples

To set the number of status enquiries that will be sent before compilation and transmission of a full status report for serial port 2 of slot 2 to 75 enquiries:

```
ssr(config)# frame-relay set lmi full-enquiry-interval 75 port se.2.2
```

frame-relay set payload-compress

Purpose

Enable packet compression for frame-relay ports.

Format

frame-relay set payload-compress [type frf9_mode1_stac] ports<*port list*>

Mode

Configure

Description

The **frame-relay set payload-compress** command allows you to enable packet compression according to Mode 1 of FRF 9. If this command is not configured, packet compression is not enabled.

Parameters

type frf9_mode1_stac
Specifies the Stacker FRF 9, Mode 1 compression algorithm. This is the default value.

<*port list*>
The port(s) on which you wish to enable the packet compression. You can specify a single VC or a comma-separated list of VCs.

Restrictions

Usage is restricted to frame relay VCs only.

Example

To enable Stacker FRF 9, Mode 1 packet compression on slot 3, VC 300 on serial port 1:

```
ssr(config)# frame-relay set payload-compress ports se.3.1.300
```

frame-relay set peer-addr

Purpose

Set the peer address in case that InArp is not supported on the remote device.

Format

frame-relay set peer-addr <*IP address*> **ports** <*port list*>

Mode

Configure

Description

Issuing the **frame-relay set peer-addr** command allows you to set the peer address if it can't be resolved by InArp.

Parameters

<*IP address*> The IP or IPX address you wish to use.

<*port list*> The location of the port to which you wish to assign the address.

Restrictions

Usage is restricted to frame relay VCs only.

Example

To assign an IP address 10.1.1.1/16 to slot 2, VC 100 on serial port 1:

```
ssr(config)# frame-relay set peer-addr ip-addr 10.1.1.1/16 ports
    se.2.1.100
```

frame-relay show service

Purpose

Displays frame relay service profiles.

Format

frame-relay show service <*service name*> | **all**

Mode

Enable

Description

The **frame-relay show service** command allows you to display the available frame relay service profiles.

Parameters

<*service name*>	The name of a particular pre-defined service profile.
all	Displays all of the available frame relay service profiles.

Restrictions

None.

Example

To display the available frame relay service profiles named “prof1”:

```
ssr# frame-relay show service prof1
```

frame-relay show stats

Purpose

Displays frame relay statistics.

Format

frame-relay show stats port <*port name*> **[last-error] [lmi] [mibII]**

Mode

Enable

Description

The **frame-relay show stats** command allows you to display the following frame relay port statistics for the given port:

- The last reported frame relay error.
- The active frame relay LMI parameters.
- The MIBII statistics for frame relay WAN ports.

Parameters

port <*port name*>
The port or ports for which you want to display statistics.

last-error
Specifying the **last-error** keyword allows you to display the last reported frame relay error for the given port.

lmi
Specifying the **lmi** keyword allows you to displays the active frame relay LMI parameters.

mibII
Specifying the **mibII** keyword allows you to displays the MIBII statistics for frame relay WAN ports.

Restrictions

The **last error**, **mibii**, and **lmi** commands are for ports only (no VC designators allowed). Otherwise, the port name may have the "VC" designator.

Examples

To display the last recorded error and MIB II statistics and for serial port 1 of slot 3:

```
ssr# frame-relay show stats port se.3.1 last-error mibII
```

To display the VC statistics for serial port 1, slot 3, VCs 1-10:

```
ssr# frame-relay show stats port se.3.1.1-10
```

frame-relay show stats summary

Purpose

Displays a summary of all VC statistics.

Format

frame-relay show stats summary port <*port name*>

Mode

Enable

Description

The **frame-relay show stats summary** command allows you to display all of the summary information for VC statistics.

Parameters

<*port name*> The port or ports for which you wish to display summary statistics.

Restrictions

None.

Example

To display summary statistics for serial port 1 of slot 4, VC 100:

```
ssr# frame-relay show stats summary port se.4.1.100
```

Chapter 18

igmp Commands

The **igmp** commands let you display and set IGMP parameters.

Command Summary

Table 12 lists the **igmp** commands. The sections following the table describe the command syntax.

Table 12. igmp commands

igmp enable interface <*name/ipAddr*>
igmp enable vlan <*vlan-name*>
igmp set interface <*name/ipAddr*> **[allowed-groups** <*group-list*> **| not-allowed-groups** <*group-list*>**] [use-all-ports]**
igmp set queryinterval <*num*>
igmp set responsetime <*num*>
igmp set vlan <*vlan-name*> **[host-timeout** <*num*>**]** **[querier-timeout** <*num*>**]** **[router-timeout** <*num*> **leave-timeout** <*num*> **]**
igmp show interfaces [group <*ipAddr*> **| interface** <*name/ipAddr*>**]**
igmp show memberships [group <*ipAddr*> **|** *port* <*num*>**]**
igmp show timers
igmp show vlans
igmp start-snooping

igmp enable interface

Purpose

Enables IGMP on an interface.

Format

igmp enable interface <*name/ipAddr*>

Mode

Configure

Description

The **igmp enable interface** command enables IGMP on the specified interface.

Parameters

<*name/ipAddr*> Name or IP address of the interface on which you are enabling IGMP.

Restrictions

IGMP is not enabled on tunnels.

Example

To enable IGMP on interface 10.50.1.2:

```
ssr(config)# igmp enable interface 10.50.1.2
```

igmp enable vlan

Purpose

Enables IGMP snooping on a VLAN.

Format

igmp enable vlan <*vlan-name*>

Mode

Configure

Description

The **igmp enable vlan** command enables IGMP snooping on a specified VLAN. By default, IGMP snooping is disabled on all VLANs.

Parameters

<*vlan-name*>
Is the name of the VLAN where IGMP snooping is to be enabled.

Restrictions

Layer 3 multicasting and layer-2 snooping cannot be run simultaneously on the same VLAN.

Example

To enable igmp snooping on VLAN blue::

```
ssr(config)# igmp enable vlan blue
```

igmp set interface

Purpose

Configures IGMP parameters.

Format

igmp set interface <*name/ipAddr*>
[allowed-groups <*group-list*>**| not-allowed-groups** <*group-list*>**] [use-all-ports]**

Mode

Configure

Description

Sets IGMP parameters on a per-interface basis to control group restrictions and optimization.

Parameters

allowed-groups <*group-list*>
Restricts the groups to only those specified.

not-allowed-groups <*group-list*>
Allows any groups besides those specified.

Note: Specify only one of the above options, as they are mutually exclusive.

use-all-ports
Disables per-port IGMP control. By default, per-port IGMP control is enabled.

Note: If the traffic is being supplied by a dvmrp tunnel, which uses CPU-based switching, then for efficiency reasons, port based optimization is not used by this traffic.

Restrictions

None.

Examples

The following is an example of the **igmp set interface** command::

```
ssr(config)# igmp set interface 200.1.1.1 allowed-groups 225.2.0.0/16
```

The above command will allow only memberships to groups falling in the specified range. Outside this range, all groups are implicitly ignored.

igmp set queryinterval

Purpose

Configures IGMP Host Membership Query interval.

Format

igmp set queryinterval <*num*>

Mode

Configure

Description

Sets the IGMP Host Membership Query time interval. The interval you set applies to all ports on the SSR.

Parameters

<*num*> A value from 20 – 3600 seconds. The default is 125 seconds.

Restrictions

None.

Example

To set the query interval to 30 seconds:

```
ssr(config)# igmp set queryinterval 30
```

igmp set responsetime

Purpose

Configures IGMP Host Membership response wait time.

Format

igmp set responsetime <*num*>

Mode

Configure

Description

Sets the wait time for IGMP Host Membership responses. The wait time you set applies to all ports on the SSR.

Parameters

<*num*> Response wait time in seconds. Specify a number from 10 – 3599. The default is 10.

Restrictions

None.

Examples

To set the Host Membership response wait time to 20 seconds:

```
ssr(config)# igmp set responsetime 20
```

igmp set vlan

Purpose

Sets parameters for IGMP snooping on a VLAN.

Format

igmp set vlan <*vlan-name*> [**host-timeout** <*num*>] [**querier-timeout** <*num*>] [**router-timeout** <*num*> **leave-timeout** <*num*>] [**filter-ports** <*port-list*>] [**permanent-ports** <*port-list*>]

Mode

Configure

Description

The **igmp set vlan** command allows you to set parameters for VLAN-based IGMP snooping.

Parameters

host-timeout <*num*>
Allows adjusting to long host timeout values that may have been set up for the IGMP querier. The default value is 250 seconds.

querier-timeout <*num*>
Allows adjusting to long timeout values that may have been set up for the IGMP querier. The default value is 260 seconds.

router-timeout <*num*>
Allows adjusting to long timeout values that may have been set up for the routers. Different versions of DVMRP can have different timeouts. The default value is 140 seconds.

leave-timeout <*num*>
Allows quicker timeout if IGMP v2 leave messages are used. The value is nominally 10 seconds.

filter-ports <*port-list*>
Allows forced filtering of certain ports from multicast data. Setting ports as filter ports

ensures that no host there will join any memberships. A port can optionally be either a permanent port or a filter port, but not both.

permanent-ports <*port-list*>
Allows forcing of mulicast data if present on certain ports. A port can optionally be either a permanent port or a filter port, but not both.

Restrictions

None.

Example

To set parameters for IGMP snooping on the VLAN blue:

```
ssr(config)# igmp set vlan blue host-timeout 125 querier-timeout 130
router-timeout 70
```

igmp show interfaces

Purpose

Shows the interfaces running IGMP.

Format

igmp show interfaces [group <*ipAddr*> | **interface** <*name/ipAddr*>**]**

Mode

Enable

Description

The **igmp show interfaces** command shows memberships on a specified interface or for a multicast group address. When you use the command to show interfaces by group, all interfaces containing the group membership are shown.

Note: This command is similar to **igmp show memberships**, except where the **igmp show interfaces** command shows interface details, the **igmp show memberships** command shows ports.

Parameters

group <*ipAddr*> Address of a multicast group.

interface <*name/ipAddr*>Name or address of a interface.

Restrictions

None.

Example

To show information about the interfaces running IGMP:

```
ssr# igmp show interfaces

Address: 172.1.1.10 Subnet: 172.1.1/24 Met: 1 Thr: 1
Name : mls15 State: Up Querier Leaf Igmp Dvmrp

Address: 207.135.89.64 Subnet: 207.135.89.0/25 Met: 1 Thr: 1
Name : company State: Up Querier Leaf Igmp Dvmrp
Groups : 224.0.1.12
224.1.127.255
224.0.1.24
224.2.127.253
224.2.127.254

Address: 10.135.89.10 Subnet: 10.135.89.0/25 Met: 1 Thr: 1
Name : test State: Up Querier Igmp Dvmrp

Address: 207.135.122.11 Subnet: 207.135.122.8/29 Met: 1 Thr: 1
Name : mbone State: Up Igmp Dvmrp
Groups : 224.0.1.11
224.0.1.12
224.2.127.254
239.255.255.255
224.2.127.253
```

igmp show memberships

Purpose

Displays IGMP host memberships.

Format

igmp show memberships [group <*ipAddr*> | *port* <*num*>**]**

Mode

Enable

Description

The **igmp show memberships** command displays IGMP host members on a specific interface and/or for a particular multicast group.

Parameters

group <*ipAddr*>	Address of the multicast group for which to display host memberships.
port <*num*>	Port numbers on which the members reside.

Restrictions

None.

Examples

To display host members for multicast group 225.0.1.20:

```
ssr(config)# igmp show memberships group 225.0.1.20
```

To display host members for multicast group 225.0.1.20 on port et.1.1:

```
ssr(config)# igmp show memberships group 225.0.1.20 port et.1.1
```

The following is a fuller example.

```
ssr(config)# igmp show memberships

Group : 224.0.1.11 Ports: et.1.1
Group : 224.0.1.12 Ports: et.1.1
et.5.1
Group : 224.0.1.24 Ports: et.5.1
Group : 224.1.127.255 Ports: et.5.1
Group : 224.2.127.253 Ports: et.1.1
et.5.1
Group : 224.2.127.254 Ports: et.1.1
et.5.1
Group : 239.255.255.255 Ports: et.1.1
```

igmp show timers

Purpose

Displays IGMP timers.

Format

igmp show timers

Mode

Enable

Description

The **igmp show timers** command displays IGMP timers.

Parameters

None.

Restrictions

None.

igmp show vlans

Purpose

Displays IGMP VLANs.

Format

igmp show vlans [detail] [name <*name*>] [timers]

Mode

Enable

Description

The **igmp show vlans** command displays IGMP VLANs.

Parameters

detail	Shows all IGMP membership information
name <*name*>	Shows IGMP membership information for the specified VLAN
timers	Shows all IGMP L2 snooping related timers

Restrictions

None.

igmp start-snooping

Purpose

Starts passive IGMP snooping on enabled VLANs.

Format

igmp start-snooping

Mode

Configure

Description

The **igmp start-snooping** command starts IGMP snooping on enabled VLANs. This task is independent of L3 multicasting.

Parameters

None.

Restrictions

None.

Chapter 19

interface Commands

The interface commands let you create IP and IPX interfaces, add network mask and broadcast address information to existing IP interfaces, and display configuration information for IP and IPX interfaces.

Command Summary

Table 13 lists the interface commands. The sections following the table describe the command syntax.

Table 13. interface commands

interface add ip <*InterfaceName*> **address-netmask** <*ipAddr-mask*> **[broadcast** <*ipaddr*>**]**
interface create ip <*InterfaceName*> **address-mask** <*ipAddr-mask*> **[broadcast** <*ipAddr*>**]** **vlan** <*name*> **|** **port** <*port*> **mtu** <*num*> **[output-mac-encapsulation** <*MACencap*>**]** **[up	down] [mac-addr** <*MACaddr-spec*>**]**
interface create ipx <*InterfaceName*> **address** <*ipxAddr*> **vlan** <*name*> **|** **port** <*port*> **[output-mac-encapsulation** <*MACencap*>**] [up	down] [mac-addr** <*MACaddr-spec*>**]**
interface show ip <*InterfaceName*> **| all**
interface show ipx <*InterfaceName*> **| all**

interface add ip

Purpose

Configure secondary addresses for an existing interface.

Format

interface add ip <*InterfaceName*> **address-mask** <*ipAddr-mask*> **[broadcast** <*ipAddr*>**]**

Mode

Configure

Description

The **interface add ip** command configures secondary addresses for an existing IP interface.

Note: The interface must already exist. To create an interface, enter the **interface create ip** command.

Parameters

<*InterfaceName*> Name of the IP interface; for example, int4.

address-netmask IP address and netmask of this interface. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the SSR uses the natural mask for the address (/8 for Class A, /16 for Class B or /24 for Class C).

broadcast <*ipAddr*> Broadcast address of this interface.

Restrictions

You can use this command only on an interface that has already been created using the **interface create ip** command.

Example

To configure a secondary address of 10.23.4.36 with a 24-bit netmask (255.255.255.0) on the IP interface int4:

```
ssr(config)# interface add ip int4 address-mask 10.23.4.36/24
```

interface create ip

Purpose

Create an IP interface.

Format

interface create ip <*InterfaceName*> **address-mask <*ipAddr-mask*> [broadcast** <*ipAddr*>**]**
vlan <*name*>**|port** <*port*> **mtu** <*num*>
[output-mac-encapsulation <*MACencap*>**] [up|down]**
[mac-addr <*MACaddr-spec*>**]**
[type broadcast|point-to-point]

Mode

Configure

Description

The **interface create ip** command creates and configures an IP interface. Configuration of an IP interface can include information such as the interface's name, IP address, netmask, broadcast address, and so on. You can also create an interface in a disabled (**down**) state instead of the default enabled (**up**) state.

The SSR is pre-allocated a pool of 64 MAC addresses. By default, each new IP interface is automatically configured with the lowest MAC address in the pool (the "base" MAC address). However, you can assign an interface a different MAC address by using the **mac-addr** option.

Interfaces on the SSR are logical interfaces. Therefore, you can associate an interface with a single port or with multiple ports.

- To associate an interface with a single port, use the **port** option with the **interface create** command.
- To associate an interface with multiple ports, first create an IP VLAN and add ports to it, then use the vlan option with the **interface create** command.

Note: You must use either the **port** option or the **vlan** option with the **interface create** command.

Parameters

<InterfaceName>
Name of the IP interface; for example, int4.

address-netmask
IP address and netmask of this interface. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the SSR uses the natural mask for the address (/8 for Class A, /16 for Class B or /24 for Class C).

vlan *<name>*
Name of the VLAN associated with this interface.

port *<port>*
Port associated with this interface.

mtu *<num>*
Sets the Maximum Transmission Unit (MTU) for this interface.

up
Sets the state of the interface to up. (This is the default state.)

down
Sets the state of the interface to down.

output-mac-encapsulation
The output MAC encapsulation associated with this interface. You can specify one of the following:

- **ethernet_ii** (the default)
- **ethernet_snap**

mac-addr *<MACaddr-spec>*
Sets the MAC address for this interface. You can specify one of the following:

- A specific MAC address – specify the entire MAC address as follows: xx:xx:xx:xx:xx:xx
- An offset from the base MAC address in the pool – specify the offset. For example, to specify an offset of 10 from the base MAC address, enter "10". For example, if the base MAC address is 00:E0:63:02:00:00 and you specify an offset of 10, the SSR assigns MAC address 00:E0:63:02:00:0A to the interface.
- The base MAC address – specify the **basemac** keyword. This is the default.

type broadcast | point-to-point
Sets the type of interface. Specify one of the following:

- **broadcast** (the default)
- **point-to-point**

Restrictions

None.

Examples

To create a VLAN called IP3, add ports et.3.1 through et.3.4 to the VLAN, then create an IP interface on the VLAN:

```
ssr(config)# vlan create IP3 ip

ssr(config)# vlan add ports et.3.1-4 to IP3

ssr(config)# interface create ip int3 address-mask 10.20.3.42/24 vlan IP3
```

To create an interface called “int7” with the address 10.50.89.88 and a 16-bit subnet mask, enter the following command. The interface is associated with port et.1.3.

```
ssr(config)# interface create ip int7 address-mask 10.50.89.88/16 port et.1.3
```

To create an interface called “int1” with a broadcast address of 10.10.42.255, enter the following command. The interface is associated with the VLAN called “marketing”. The interface is created in the down (disabled) state.

```
ssr(config)# interface create ip int1 address-mask 10.10.42.17/255.255.255.0
  broadcast 10.10.42.255 vlan marketing down
```

interface create ipx

Purpose

Create an IPX interface.

Format

interface create ipx <*InterfaceName*> **address** <*ipxAddr*>
vlan <*name*> | **port** <*port*>
[output-mac-encapsulation <*MACencap*>**] [up** | **down]**
[mac-addr <*MACaddr-spec*>**]**

Mode

Configure

Description

The **interface create ipx** command creates and configures an IPX interface. Configuration of an IPX interface can include information such as the interface's name, IPX address, VLAN, port, and output MAC encapsulation. You can also create an interface in the disabled (**down**) state instead of the default enabled (**up**) state.

The SSR is pre-allocated a pool of 64 MAC addresses. By default, each new IPX interface is automatically configured with the lowest MAC address in the pool (the "base" MAC address). However, you can assign an interface a different MAC address by using the **mac-addr** option.

Parameters

<*InterfaceName*>
Name of the IPX interface; for example, int9.

address <*ipxAddr*>
IPX address of this interface.

vlan <*name*>
Name of the VLAN associated with this interface.

port <*port*>
Port associated with this interface.

up
Sets the state of the interface to up. (This is the default state.)

down
Sets the state of the interface to down.

output-mac-encapsulation
The output MAC encapsulation associated with this interface. You can specify one of the following:

- **ethernet_ii** (the default)
- **ethernet_snap**
- **ethernet_802.2_ipx**

mac-addr <*MACaddr-spec*>
Sets the MAC address for this interface. You can specify one of the following:

- A specific MAC address – specify the entire MAC address as follows: xx:xx:xx:xx:xx:xx
- An offset from the base MAC address in the pool – specify the offset. For example, to specify an offset of 10 from the base MAC address, enter "10". For example, if the base MAC address is 00:E0:63:02:00:00 and you specify an offset of 10, the SSR assigns MAC address 00:E0:63:02:00:0A to the interface.
- The base MAC address – specify the **basemac** keyword. This is the default.

Restrictions

None.

Examples

The following commands create a VLAN called IPX10, add all the ports on the line card in slot 1 to the VLAN, and create an IPX interface called "int10" with the IPX address a98d7c6f, associated with VLAN IPX10.

```
ssr(config)# vlan create IPX10 ipx

ssr(config)# vlan add ports et.1.* to IPX10

ssr(config)# interface create ipx int10 address a98d7c6f vlan IPX10
```

The following command creates an interface called “int5” with the IPX address 82af3d57 for port et.1.3. The interface is added in the down (disabled) state.

```
ssr(config)# interface create ipx int5 address 82af3d57 port et.1.3 down
```

To create an interface called “int6” with the MAC address 00:01:02:03:04:05 and IPX address 82af3d58 for port et.1.4.

```
ssr(config)# interface create ipx int6 address 82af3d58 port et.1.4
  mac-addr 00:01:02:03:04:05
```

To create an interface called “int7” for a VLAN called “IPX-VLAN” on port et.1.4 with the MAC address at the base of the SSR’s MAC address pool:

```
ssr(config)# interface create ipx int7 address 82af3d59 vlan IPX-VLAN et.1.4
  mac-addr basemac
```

The following command creates an interface called “int7” for a VLAN called “IPX-VLAN” on port et.1.4 with a MAC address offset by 10 from the base of the SSR’s MAC address pool. If the base MAC address in the SSR’s MAC address pool is 00:E0:63:02:00:00, the offset of 10 gives the interface the MAC address 00:E0:63:02:00:0A.

```
ssr(config)# interface create ipx int7 address 82af3d59 vlan IPX-VLAN et.1.4
  mac-addr 10
```

interface show ip

Purpose

Display configuration of an IP interface.

Format

interface show ip <*InterfaceName*> | **all**

Mode

Enable

Description

The **interface show ip** command displays configuration information for an IP interface.

Note: You can display exactly the same information from within the ip facility using the **ip show interfaces** command.

Parameters

<*InterfaceName*> | **all**
Name of the IP interface; for example, int4. Specify **all** to show configuration information about all the IP interfaces on the SSR.

Restrictions

None.

Examples

To display configuration information for the IP interface called "int7":

```
ssr# interface show ip int7
```

.To display configuration information for all IP interfaces:

```
ssr# interface show ip all
```

interface show ipx

Purpose

Display configuration of an IPX interface.

Format

interface show ipx <*InterfaceName*> | **all**

Mode

Enable

Description

The **interface show ipx** command displays configuration information for an IPX interface.

Note: You can display exactly the same information from within the ip facility using the **ipx show interfaces** command.

Parameters

<*InterfaceName*> | **all**
Name of the IPX interface; for example, int9. Specify **all** to show configuration information about all the IPX interfaces on the SSR.

Restrictions

None.

Examples

To display configuration information for the IPX interface called "int8":

```
ssr# interface show ipx int8
```

To display configuration information for all IPX interfaces:

```
ssr# interface show ipx all
```

Chapter 20

ip Commands

The **ip** commands let you display route table entries and various IP related tables.

Command Summary

Table 14 lists the **ip** commands. The sections following the table describe the command syntax.

Table 14. ip commands

ip add route <*ipAddr-mask*> | **default gateway** <*hostname-or-IPaddr*> **[host]** **[interface** <*hostname-or-IPaddr*>**]** **[preference** <*num*>**]** **[retain]** **[reject]** **[no-install]** **[blackhole]** **[gate-list** <*gateway list*>**]**
ip disable deny-attack | dns-lookup | fast-icmp | forwarding | **[icmp-redirect interface** <*name*> **| all] | [proxy-arp interface** <*name*> **| all] |** **source-routing**
ip dos disable port-attack-protection | directed-broadcast-protection
ip enable directed-broadcast
ip helper-address interface <*interface-name*> <*helper-address*> **| all-interfaces** [<*udp-port#*>]
ip l3-hash channel <*num*> **| all variant** <*num*>
ip set data-receive-size | control-receive-size <*num*>
ip set port <*port-list*> **forwarding-mode destination-based**
ip show connections [no-lookup]

Table 14. ip commands (Continued)

ip show helper-address
ip show interfaces [<*interface-name*>]
ip show routes [no-lookup] [show-arps] [show-multicast] [verbose]

ip add route

Purpose

Configure a static route.

Format

ip add route <*ipAddr-mask*> | **default gateway** <*hostname-or-IPaddr*> **[host] [interface** <*hostname-or-IPaddr*>**] [preference** <*num*>**] [retain] [reject] [no-install] [blackhole] [gate-list** <*gateway list*>**]**

Mode

Configure

Description

The **ip add route** command creates a static route entry in the route table. The static route can be a default route, a route to a network, or a route to a specific host.

Parameters

<*ipAddr-mask*> IP address and netmask of the destination. You can specify the address and mask information using the traditional format (example: 10.1.2.3/255.255.0.0) or the CIDR format (example: 10.1.2.3/16). If you specify an address without mask information, the SSR uses the natural mask for the address (/8 for Class A, /16 for Class B or /24 for Class C).

gateway <*hostname-or-IPaddr*>
IP address or hostname of the next hop router for this route.

host Specifies that this route is a route to a host.

interface The next hop interface associated with this route. When this option is specified, gateways are only considered valid when they are on one of these interfaces

preference The preference of this static route. The preference controls how this route competes with routes from other protocols. The parameter takes a value between 0-255. The default preference is 60.

retain If specified, this option prevents this static route from being removed from the forwarding table when the routing service (GateD) is gracefully shutdown. Normally gated removes all routes except interface routes during a graceful shutdown. The retain option can be used to insure that some routing is available even when GateD is not running.

reject If specified, install this route as a reject route. Instead of forwarding a packet like a normal route, reject routes cause packets to be dropped and unreachable messages to be sent to the originator of the packet.

no-install If specified, the route will not be installed in the forwarding table when it is active but will be eligible for exporting to other protocols.

blackhole This option is the same as the reject option with the exception that unreachable messages are not sent.

gate-list <*gateway list*>
Allows you to specify up to four gateways for a particular destination host or network.

Restrictions

None

Examples

To configure the router 10.4.1.1 as the default gateway for this SSR:

```
ssr(config)# ip add route default gateway 10.4.1.1
```

To configure the gateway 10.4.78.11 as the gateway for any packet destined for the subnet 10.4.14.0/24:

```
ssr(config)# ip add route 10.4.14.0/24 gateway 10.4.78.11
```

To configure the gateway 10.4.78.11 as the gateway for any packet destined for the subnet 10.4.14.0/24:

```
ssr(config)# ip add route 10.4.14.0/24 gateway 10.4.78.11
```

To configure the gateway 10.4.16.99 as the gateway to the host 10.4.15.2:

```
ssr(config)# ip add route 10.4.15.2 host gateway 10.4.16.99
```

To configure a reject route entry for packets destined for the subnet 10.14.3.0/24:

```
ssr(config)# ip add route 10.14.3.0/24 gateway 10.1.16.99 reject
```

ip disable

Purpose

Disables IP options on the SSR.

Format

ip disable dns-lookup | fast-icmp | forwarding |
[icmp-redirect interface <*name*> **| all] | [proxy-arp interface** <*name*> **| all] |**
source-routing

Mode

Configure

Description

The **ip disable** command allows you to disable features that are enabled by default on the SSR.

Parameters

dns-lookup
Disables DNS name lookup for all commands. Sometimes a DNS server is too slow to respond and this can cause a command that displays information about many hosts to take a long time to finish. Disabling DNS lookup displays all host addresses as IP addresses instead of host names.

fast-icmp
Disables the fast ICMP feature on the SSR. By default, the SSR installs ICMP flows to be switched along the fast path in hardware if the ICMP flow is meant to be routed. ICMP echo requests are installed as control priority for packets destined for the SSR. When this feature is disabled, all ICMP packets are handled via the slow path in software.

forwarding
Disables the router's ability to forward IP packets. No IP packets will be forwarded to any IP interface if this command is used.

icmp-redirect interface <*interface name*> | **all**
Disables ICMP redirection on the specified IP interface. If you specify the all keyword, ICMP redirection is disabled for all network interfaces.

proxy-arp interface <*name*> | **all**
Disables the proxy ARP feature on the specified IP interface. By default, the **SSR** acts as a proxy for ARP requests with destination addresses of hosts to which the **SSR** can route traffic. Unless you actually require the use of proxy ARP, it is advisable to disable it on the **SSR**. If you specify the all keyword, the proxy ARP feature is disabled for all network interfaces.

source-routing
Causes the **SSR** to drop packets that have the SOURCE_ROUTE option set in the IP header. By default, packets that have the SOURCE_ROUTE option set are forwarded using the next-hop address in the IP packet.

Restrictions

None

Examples

To disable ICMP redirection on the "int4" network interface:

```
ssr(config)# ip disable icmp-redirect int4
```

To disable DNS name lookup for all commands:

```
ssr(config)# ip disable icmp-redirect dns-lookup
```

To prevent the SSR from acting as a proxy for ARP requests with destination addresses of hosts to which the SSR can route traffic:

```
ssr(config)# ip disable proxy-arp interface all
```

ip dos disable

Purpose

Disables denial of service (DOS) features on the SSR.

Format

ip dos disable directed-broadcast-protection | port-attack-protection

Mode

Configure

Description

By default, the SSR 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 behavior with the **ip dos disable directed-broadcast-protection** command.

Similarly, the SSR installs flows to drop packets destined for the SSR for which service is not provided by the SSR. This prevents packets for unknown services from slowing the CPU. You can disable this behavior with the **ip dos disable port-attack-protection** command, causing these packets to be processed by the CPU.

Parameters

directed-broadcast-protection
Disables the directed-broadcast-protection feature of the SSR. By default the SSR drops packets sent as directed broadcasts if directed broadcast is not enabled on the interface where the packet is recieved. This command causes directed broadcast packets to be processed on the SSR even if directed broadcast is not enabled on the interface receiving the packet.

port-attack-protection
Disables the port-attack-protection feature of the SSR. By default, packets that are destined for the SSR, but do not have a service defined for them on the SSR, are dropped. This prevents packets for unknown services from slowing the SSR's CPU. This command disables this behavior, allowing packets destined for the SSR that do not have a service defined for them on the SSR to be processed by the SSR's CPU.

Restrictions

None

Examples

To cause directed broadcast packets to be processed on the SSR, even if directed broadcast is not enabled on the interface receiving the packet:

```
ssr(config)# ip dos disable directed-broadcast-protection
```

To allow packets destined for the SSR, but do not have a service defined for them on the SSR, to be processed by the SSR's CPU:

```
ssr(config)# ip dos disable port-attack-protection
```

ip enable directed-broadcast

Purpose

Configure the router to forward directed broadcast packets received on an interface.

Format

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

Mode

Configure

Description

Directed broadcast packets are network or subnet broadcast packets which are sent to a router to be forwarded as broadcast packets. They can be misused to create Denial Of Service attacks. The SSR protects against this possibility by ***not*** forwarding directed broadcasts, by default. To enable the forwarding of directed broadcasts, use the **ip enable directed-broadcast** command.

Parameters

interface <*interface name*> | **all**
This is the name of the specified IP interface. If you specify the **all** keyword, directed broadcast forwarding is enabled for all network interfaces.

Restrictions

None

Examples

To enable directed broadcast forwarding on the “int4” network interface:

```
ssr(config)# ip enable directed-broadcast interface int4
```

To enable directed broadcast forwarding for all network interfaces:

```
ssr(config)# ip enable directed-broadcast interface all
```

ip helper-address

Purpose

Configure the router to forward specific UDP broadcast packets across interfaces.

Format

ip helper-address interface <*interface-name*> <*helper-address*> | **all-interfaces** [<*udp-port#*>]

Mode

Configure

Description

The **ip helper-address** command allows the user to forward specific UDP broadcast from one interface to another. Typically, broadcast packets from one interface are not forwarded (routed) to another interface. However, some applications use UDP broadcast to detect the availability of a service. Other services, for example BOOTP/DHCP require broadcast packets to be routed so that they can provide services to clients on another subnet. An IP helper can be configured on each interface to have UDP broadcast packets forwarded to a specific host for a specific service or forwarded to all other interfaces.

The **ip helper-address** command allows the user to 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 SSR 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)

Parameters

<interface-name> Name of the IP interface where UDP broadcast is to be forwarded to the helper address.

<helper-address> | **all-interfaces**
Address of the host where UDP broadcast packets should be forwarded. If **all-interfaces** is specified, UDP broadcast packets are forwarded to all interfaces except the interface on which the broadcast packet was received.

<udp-port> Destination UDP port number of the broadcast packets to forward. If not specified, packets for the six default services will be forwarded to the helper address.

Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

Examples

To forward UDP broadcast packets received on interface int1 to the host 10.1.4.5 for the six default UDP services:

```
ssr(config)# ip helper-address interface int1 10.1.4.5
```

To forward UDP broadcast packets received on interface int2 to the host 10.2.48.8 for packets with the destination port 111 (port mapper):

```
ssr(config)# ip helper-address interface int2 10.2.48.8 111
```

To forward UDP broadcast packets received on interface int3 to all other interfaces:

```
ssr(config)# ip helper-address interface int3 all-interfaces
```

ip l3-hash

Purpose

Changes the hashing algorithm used for the L3 lookup table.

Format

ip l3-hash channel <*num*> | **all variant** <*num*>

Mode

Configure

Description

The SSR's L3 Lookup table is organized as a hash table. The hash function reduces the destination and source MAC addresses to 16-bit quantities each. The hashing algorithm generates a uniform distribution within the MAC address space. However, given a particular set of addresses, the distribution may cause addresses to clump together in the table. To minimize the risk of thrashing in the tables, three variations to the basic hashing algorithm are defined. Only one variation is in effect on a line card at any given time. You can use the ip **l3-hash** command to control which variation is in effect for a line card.

To see the effect changing the hashing algorithm has on the hash bucket, use the **statistics show l3-stat** command in the SSR's Diag mode.

Parameters

channel <*num*> | **all**
Is a slot number on the SSR. Valid slot numbers are 0-3 on the SSR 2000, 0-7 on the SSR 8000, and 0-15 on the SSR 8600. The hashing algorithm change affects all ports on the line card in the slot. The **all** option causes the hashing algorithm to change on all ports on all slits.

variant <*num*>
Causes a variation to the basic hashing algorithm to be made. Valid variant numbers are 1-3. If you specify 0, the default hashing algorithm is used.

Restrictions

None.

Example

To change the default hashing algorithm used for the L3 lookup table on all ports on slot 7:

```
ssr(config)# ip l3-hash channel 7 variant 1
```

ip set data-receive-size | control-receive-size

Purpose

Sets the size of the stack data and control receive queues.

Format

ip set data-receive-size | control-receive-size <*num*>

Mode

Configure

Description

The **ip set data-receive-size | control-receive-size** command allows you to tune the size of the data and control pipes that reside between the IP stack and internal drivers on the Control Module.

Parameters

data-receive-size <*num*>
Sets the size of the stack data receive queue. Specify a value from 256-1024 bytes. The default is 512 bytes.

control-receive-size <*num*>
Sets the size of the stack control receive queue. Specify a value from 256-1024 bytes. The default is 512 bytes.

Restrictions

None.

Example

To set the size of the stack data receive queue to 1024 bytes:

```
ssr(config)# ip set data-receive-size 1024
```

ip set port forwarding-mode

Purpose

Causes the SSR, when processing an IP packet, to extract only certain fields from a layer-4 flow, rather than the entire flow.

Format

ip set port <*port-list*> **forwarding-mode <destination-based | host-flow-based>**

Mode

Configure

Description

The SSR's flow identifying logic normally extracts the complete application (layer-4) flow from an IP packet. The **ip set port forwarding-mode** command causes the SSR to extract only certain flow-related fields from the packet's L3 header, rather than the full layer-4 flow. This allows ports to route packets based on destination address alone, or on destination and source address only. As a result, in environments that do not have any filtering or RSVP requirements, the flow table can be used much more efficiently.

Parameters

port <*port-list*>
Modifies the flow extraction behavior on the specified ports. All ports must have an IP interface configured for them.

destination-based
If the packet is a unicast packet, causes the *destination IP address*, *TOS* and *L4 protocol* fields to be the only fields extracted from the IP packet. These fields and the *port of entry* field are set into the flow block being constructed. All of the other fields are set to zero.

For L3 multicast packets, the *destination IP address*, *source IP address, TOS* and *L4 protocol* fields are the only fields extracted from the IP packet. These along with the *port of entry* are the only fields set in the flow block. The remaining fields are set to zero. The flow lookup then proceeds as normal.

host-flow-based

For both unicast and multicast packets, the *destination IP address*, *source IP address*, *TOS* and the *L4 protocol* are the only fields extracted from the IP packet. These along with the *port of entry* are set in the flow block. The remaining flow block fields are set to zero. The flow lookup then proceeds as normal.

Restrictions

None

Example

To cause the SSR to extract only the *destination IP address*, *TOS*, and *L4 protocol* fields from a layer-4 flow when processing an IP packet on port et.1.1:

```
ssr(config)# ip set port et.1.1 forwarding-mode destination-based
```

To cause the SSR to extract only the *destination IP address*, *source IP address*, *TOS*, and *L4 protocol* type from a layer-4 flow when processing an IP packet on port et.1.1:

```
ssr(config)# ip set port et.1.1 forwarding-mode host-flow-based
```

ip show connections

Purpose

Show all TCP/UDP connections and services.

Format

ip show connections [no-lookup]

Mode

Enable

Description

The **ip show connections** command displays all existing TCP and UDP connections to the SSR as well as TCP/UDP services available on the SSR.

Parameters

no-lookup By default, when displaying an IP address, this command attempts to do a reverse DNS lookup to look for the hostname associated with the IP address and display the hostname instead. If you do not want the reverse DNS lookup to occur, specify the **no-lookup** option.

Restrictions

None.

Example

The following example displays all established connections and services of the SSR.

```
ssr# ip show connections
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address
(state)
tcp        0      0  *:gated-gii           *:*                    LISTEN
tcp        0      0  *:http                *:*                    LISTEN
tcp        0      0  *:telnet              *:*                    LISTEN
udp        0      0  127.0.0.1:1025        127.0.0.1:162
udp        0      0  *:snmp                 *:*
udp        0      0  *:snmp-trap            *:*
udp        0      0  *:bootp-relay          *:*
udp        0      0  *:route                *:*
udp        0      0  *:*                    *:*
```

ip show helper-address

Purpose

Display the configuration of IP helper addresses.

Format

ip show helper-address [<*interface-name*>**]**

Mode

Enable

Description

The **ip show helper-address** command displays the configuration of IP helper addresses configured on the system. One can specify the optional parameter, *interface-name*, to show only the IP helper addresses configured for that interface. If the command is executed without specifying an interface name then the IP helper address configuration of all interfaces are shown.

Parameters

<*interface-name*> Name of the IP interface to display any configured IP helper addresses.

Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

Example

The following example shows that interface int4 has one helper address configured while interface int3 has one helper address configured for the port mapper service (port 111).

```
ssr# ip show helper-address
Interface       IP address      Helper Address
---------       ----------      --------------
int6            10.1.17.1       none
int5            10.1.16.1       none
int4            10.1.15.1       10.4.1.45
int1            10.1.12.1       none
int0            10.1.11.1       none
int3            10.1.14.1       10.5.78.122(111)
```

ip show interfaces

Purpose

Display the configuration of IP interfaces.

Format

ip show interfaces [<*interface-name*>**]**

Mode

Enable

Description

The **ip show interfaces** command displays the configuration of an IP interface. If you issue the command without specifying an interface name then the configuration of all IP interfaces is displayed. This command displays the same information as the **interface show ip** command.

Parameters

<*interface-name*> Name of the IP interface; for example, ssr4. If you do not specify an interface name, the SSR displays all the IP interfaces.

Restrictions

If you specify an interface name, the name must belong to an existing IP interface.

Example

To display the configuration of the IP interface "int1":

```
ssr# ip show interfaces int1
int1: flags=9862<BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: IP2
        Ports:
        inet 10.1.12.1/24 broadcast 10.1.12.255
```

ip show routes

Purpose

Display the IP routing table.

Format

ip show routes [no-lookup] [show-arps] [show-multicast] [verbose]

Mode

Enable

Description

The **ip show routes** command displays the IP routing table. Different command options can be used to show different aspects of the routing table.

Parameters

no-lookup	By default, when displaying an IP address, this command attempts to do a reverse DNS lookup to look for the hostname associated with the IP address and display the hostname instead. If you do not want the reverse DNS lookup to occur, specify the **no-lookup** option.
show-arps	By default, ARP entries are not shown. To show ARP entries (if any are present), specify the **show-arps** option.
show-multicast	By default, routes to multicast destinations are not shown. To show routes to multicast destinations, specify the **show-multicast** option.
verbose	Show the routing table in verbose mode. The additional information is useful for debugging.

Restrictions

None.

Example

The following example displays the contents of the routing table. It shows that some of the route entries are for locally connected interfaces ("directly connected"), while some of the other routes are learned from RIP.

```
ssr# ip show routes
Destination                Gateway                    Owner     Netif
-----------                -------                    -----     -----
10.1.0.0/16                50.1.1.2                   RIP       to-linux2
10.2.0.0/16                50.1.1.2                   RIP       to-linux2
10.3.0.0/16                50.1.1.2                   RIP       to-linux2
10.4.0.0/16                50.1.1.2                   RIP       to-linux2
14.3.2.1                   61.1.4.32                  Static    int61
21.0.0.0/8                 50.1.1.2                   RIP       to-linux2
30.1.0.0/16                directly connected         -         to-goya
50.1.0.0/16                directly connected         -         to-linux2
61.1.0.0/16                directly connected         -         int61
62.1.0.0/16                50.1.1.2                   RIP       to-linux2
68.1.0.0/16                directly connected         -         int68
69.1.0.0/16                50.1.1.2                   RIP       to-linux2
127.0.0.0/8                127.0.0.1                  Static    lo
127.0.0.1                  127.0.0.1                  -         lo
210.11.99.0/24             directly connected         -         int41
```

Chapter 21

ip-policy Commands

The **ip-policy** commands let you set up policies that cause the SSR to forward packets to a specified IP address based on information in a packet's L3/L4 IP header fields.

Command Summary

Table 15 lists the **ip-policy** commands. The sections following the table describe the command syntax.

Table 15. ip-policy commands

ip-policy <*name*> **apply local** | **interface** <*name*> | **all**
ip-policy clear all | **policy-name** <*name*> | **all**
ip-policy <*name*> **deny acl** <*aclname*> | **everything-else** [**sequence** <*num*>]
ip-policy <*name*> **permit acl** <*aclname*> | **everything-else** [**sequence** <*num*>] **next-hop-list** <*ip-addr-list*> **action policy-first** | **policy-last** | **policy-only**
ip-policy <*name*> **set [pinger on] [load-policy first-available** | **round-robin** | **ip-hash sip** | **dip** | **both]**
ip-policy show [all] [policy-name <*name*> | **all] [interface** <*name*> | **all]**

ip-policy apply

Purpose

Applies an IP policy to an interface.

Format

ip-policy <*name*> **apply local** | **interface** <*InterfaceName*> | **all**

Mode

Configure

Description

Once you have defined an IP policy, you use the **ip-policy apply** command to apply the IP policy to an interface. Once the IP policy is applied to the interface, packets start being forwarded using the policy.

Parameters

<*name*>	Is the name of a previously defined IP policy.
<*InterfaceName*>	Is the name of the inbound interface to which you are applying the IP policy.
local	Causes packets generated by the SSR to be forwarded according to the IP policy.
all	Causes the IP policy to be applied to all IP interfaces.

Restrictions

IP policies can be applied to IP interfaces only.

Examples

To apply IP policy p1 to interface int4:

```
ssr(config)# ip-policy p1 apply interface int4
```

To apply IP policy p2 to all IP packets generated on the SSR:

```
ssr(config)# ip-policy p2 apply local
```

ip-policy clear

Purpose

Clears IP policy statistics.

Format

ip-policy clear all | policy-name <*name*> **| all**

Mode

Enable

Description

The **ip-policy clear** command is used in conjunction with the **ip-policy show** command, which gathers statistics about IP policies. The **ip-policy clear** command lets you reset IP policy statistics to zero.

Parameters

<*name*> Is the name of an active IP policy.

all Causes statistics to be cleared for all IP policies.

Restrictions

None.

Examples

To clear statistics for IP policy p1:

```
ssr# ip-policy clear policy-name p1
```

To clear statistics for all IP policies:

```
ssr(config)# ip-policy clear all
```

ip-policy deny

Purpose

Specifies which packets cannot be subject to policy-based routing.

Format

ip-policy <*name*> **deny acl** <*aclname*> | **everything-else** [**sequence** <*num*>]

Mode

Configure

Description

The **ip-policy deny** command allows you to specifically prevent packets matching a profile from being forwarded with an IP policy. These packets are routed using dynamic routes instead.

Note: Since there is an implicit deny rule at the end of all IP policies, all packets that do not match any policy are forwarded using dynamic routes.

Parameters

<*name*>
Is the name of an IP policy.

acl <*aclname*>
Is the name of the ACL profile of the packets to be excluded from IP policy-based forwarding. Profiles are defined with the **acl** command. The ACL may contain either **permit** or **deny** keywords. The **ip-policy deny** command only looks at the following ACL rule parameter values: protocol, source IP address, destination IP address, source port, destination port, and TOS.

everything-else
Keyword that specifies an action to be performed for packets that do not match any of the previously-defined ACLs. Specifies that packets that are not ***specifically*** permitted to use policy-based routing are forwarded using dynamic routes.

sequence <*num*>
If an IP policy is composed of more than one **ip-policy** statement, specifies the order

in which the statement is evaluated. Possible values are 1-65535. The **ip-policy** statement with the lowest sequence number is evaluated first.

Restrictions

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

Examples

To create a profile called "prof1" for telnet packets from 9.1.1.5 to 15.1.1.2:

```
ssr(config)# acl prof1 permit ip 9.1.1.5 15.1.1.2 any any telnet 0
```

Note: See "acl permit | deny ip" on page 45 for more information on creating profiles for IP policy routing.

To create an IP policy called "p3" that prevents packets matching prof1 (that is, telnet packets from 9.1.1.5 to 15.1.1.2) from being forwarded using an IP policy:

```
ssr(config)# ip-policy p3 deny acl prof1
```

To create a policy called "p4" that prevents all packets that have not been specifically permitted to use policy-based routing (using the **ip-policy permit** command) from being forwarded using an IP policy:

```
ssr(config)# ip-policy p4 deny acl everything-else
```

ip-policy permit

Purpose

Specifies gateways and actions for IP policies

Format

ip-policy <*name*> **permit acl** <*aclname*> | **everything-else** [**sequence** <*num*>] [**next-hop-list** <*ip-addr-list*> | **null**] [**action policy-first** | **policy-last** | **policy-only**]

Mode

Configure

Description

The **ip-policy permit** command allows you to specify the next-hop gateway where packets matching a given profile should be forwarded. You can specify up to four next-hop gateways for an IP policy. Packets matching a profile you defined with an **acl** command are forwarded to the next-hop gateway.

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

Parameters

<*name*>
Is the name of an IP policy.

acl <*aclname*>
Is the name of the ACL profile of the packets to be forwarded using an IP policy. Profiles are created with the **acl** command. The ACL may contain either **permit** or **deny** keywords. The **ip-policy permit** command only looks at the following ACL rule parameter values: protocol, source IP address, destination IP address, source port, destination port, and TOS.

everything-else
Specifies that all packets not ***specified*** using policy-based routing (i.e., with the **ip-policy deny** command) are forwarded to the next-hop gateway.

sequence <*num*>
If an IP policy is composed of more than one **ip-policy** statement, specifies the order in which the statement is evaluated. Possible values are 1-65536. The **ip-policy** statement with the lowest sequence number is evaluated first.

next-hop-list <*ip-addr-list*> | **null**
Is the IP address of one or more next-hop gateways. Packets matching the profile specified in <*aclname*> are forwarded to one of the gateways specified here. You can specify up to four gateways for each profile. If you specify more than one gateway, enclose the list of IP addresses in quotes. You can define how the packet load is distributed among multiple gateways with the **ip-policy set load-policy** command.

To drop packets that match the profile, use the **null** keyword.

action policy-first | policy-last | policy-only
Specifies how IP policies are applied with respect to dynamic or statically configured routes. The following options are available:

policy-first Causes packets matching the specified profile to use the IP policy route first. If the next-hop gateway specified in the IP policy is not reachable, the dynamic route is used instead.

policy-last Causes packets matching the specified profile to be routed using dynamic routes first. If a dynamic route is not available, then all packets matching the profile are routed using the IP policy gateway.

policy-only Causes packets matching the specified profile to use the IP policy route. If the next-hop gateway specified in the IP policy is not reachable, then the packets are dropped.

Restrictions

ACLs for non IP protocols cannot be used for IP policy routing.

Examples

To create a profile called "prof1" for telnet packets from 9.1.1.5 to 15.1.1.2:

```
ssr(config)# acl prof1 permit ip 9.1.1.5 15.1.1.2 any any telnet 0
```

Note: See "acl permit | deny ip" on page 45 for more information on creating profiles for IP policy routing.

To cause packets matching prof1 (that is, telnet packets from 9.1.1.5 to 15.1.1.2) to be forwarded to 10.10.10.10:

```
ssr(config)# ip-policy p5 permit acl prof1 next-hop-list 10.10.10.10
```

To cause all packets that have not been specified using policy-based routing (using the **ip-policy deny** command) to be forwarded to 10.10.10.10:

```
ssr(config)# ip-policy p5 permit acl everything-else next-hop-list
10.10.10.10
```

To cause packets matching prof1 to use dynamic routes if 10.10.10.10 is not available:

```
ssr(config)# ip-policy p5 permit acl prof1 next-hop-list 10.10.10.10
action policy-first
```

To cause packets matching prof1 to be dropped if 10.10.10.10 is not available:

```
ssr(config)# ip-policy p5 permit acl prof1 next-hop-list 10.10.10.10
action policy-only
```

ip-policy set

Purpose

Controls how packets are distributed among the next hop gateways in an IP policy and queries the availability of next-hop gateways.

Format

ip-policy <*name*> **set [pinger on] [load-policy first-available | round-robin | ip-hash sip | dip | both]**

Mode

Configure

Description

If you specify more than one next-hop gateway in an IP policy, you can use the **ip-policy set** command to control how the load is distributed among the next-hop gateways. 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. You can also control which information in the IP packet to use to determine the next-hop gateway.

In addition, you can use the **ip-policy set** command to have the SSR query the availability of the next-hop gateways specified in an IP policy. When this option is active, the SSR periodically queries the next-hop gateways via ICMP_ECHO_REQUESTS. Only gateways that respond to these requests are used for forwarding packets.

Parameters

<*name*>
Is the name of an IP policy.

pinger on
Causes the SSR to 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.

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.

load-policy first-available | round-robin
If an IP policy has more than one next-hop gateway, specifies how the packets are distributed among the gateways. Two options are available:

first-available Uses the first available next-hop gateway in the **ip-policy permit** statement for all flows. This is the default.

round-robin Sequentially picks the next gateway in the list for each new flow.

load-policy ip-hash sip | dip | both
Specifies which information in the IP packet to use to determine the next hop gateway.

sip Uses the source IP based selection.

dip Uses the destination IP based selection.

both Uses both source IP and destination IP for selection.

Restrictions

None.

Examples

To set up 10.10.10.10 and 10.10.10.5 as next-hop gateways for IP policy p6:

```
ssr(config)# ip-policy p6 permit profile prof1 next-hop-list
‘10.10.10.10 10.10.10.5’
```

To distribute flows among these two next-hop gateways in a sequential manner:

```
ssr(config)# ip-policy p6 set load-policy round-robin
```

ip-policy show

Purpose

Displays information about active IP policies.

Format

ip-policy show [all] [policy-name <*name*> **| all] [interface** <*name*> **| all]**

Mode

Enable

Description

The **ip-policy show** command displays 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.

Parameters

policy-name <*name*> **| all**
Is the name of an IP policy. Use the **all** keyword to display all active policies.

Note: The **ip-policy show all** command works identically to the **ip-policy show policy-name all** command

interface <*name*> **| all**
Displays information about IP policies that have been applied to a specified interface. If you use the **all** keyword, the command displays information about IP policies that have been applied to all interfaces (that is, by using the **ip-policy apply interface all** command).

Restrictions

None.

Example

To display information about IP policy p1:

```
ssr# 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 exaluated last, causing all packets that do not match one of the profiles to be forwarded normally (with dynamic routes).

Chapter 22

ip-router Commands

The **ip-router** commands let you configure and monitor features and functions that work across the various routing protocols.

Command Summary

Table 16 lists the **ip-router** commands. The sections following the table describe the command syntax.

Table 16. ip-router commands

ip-router authentication add key-chain <*option-list*>
ip-router authentication create key-chain <*option-list*>
ip-router find route <*ip-addr*>
ip-router global add <*option-list*>
ip-router global set <*option-list*>
ip-router global set trace-options <*option-list*>
ip-router global set trace-state on | **off**
ip-router global use provided_config
ip-router kernel trace <*option-list*> **detail** | **send** | **receive**
ip-router policy add filter <*option-list*>
ip-router policy add optional-attributes-list <*option-list*>
ip-router policy aggr-gen destination <*name*> <*option-list*>
ip-router policy create aggregate-export-source <*option-list*>

Table 16. ip-router commands (Continued)

ip-router policy create aggr-gen-dest <*option-list*>
ip-router policy create aggr-gen-source <*option-list*>
ip-router policy create aspath-export-source <*number-or-string*> <*option-list*>
ip-router policy create bgp-export-destination <*number-or-string*> <*option-list*>
ip-router policy create bgp-export-source <*number-or-string*> <*option-list*>
ip-router policy create bgp-import-source <*number-or-string*> <*option-list*>
ip-router policy create direct-export-source <*option-list*>
ip-router policy create filter <*option-list*>
ip-router policy create optional-attributes-list <*option-list*>
ip-router policy create ospf-export-destination <*number-or-string*> <*option-list*>
ip-router policy create ospf-export-source <*number-or-string*> <*option-list*>
ip-router policy create ospf-import-source <*number-or-string*> <*option-list*>
ip-router policy create rip-export-destination <*number-or-string*> <*option-list*>
ip-router policy create rip-export-source <*number-or-string*> <*option-list*>
ip-router policy create rip-import-source <*number-or-string*> <*option-list*>
ip-router policy create static-export-source <*option-list*>
ip-router policy create tag-export-source <*number-or-string*> <*option-list*>
ip-router policy export destination <*option-list*>
ip-router policy import source <*option-list*>
ip-router policy redistribute from-proto <*protocol*> <*option-list*> **to-proto rip | ospf | bgp**
ip-router show configuration-file active | permanent
ip-router show rib [detail]
ip-router show route [ip-addr-mask | default] [detail]
ip-router show state [all] [memory] [timers] [to-file] [to-terminal] [task <*string*> | all | gii | icmp | inet | interface | krt | route]

ip-router authentication add key-chain

Purpose

Add a key to an existing key-chain.

Format

ip-router authentication add key-chain <*option-list*>

Mode

Configure

Parameters

<*option-list*>
Specifies the options you are adding. Specify one of the following:

key <*string*>
Adds a new key to an existing key-chain. The key can be up to 16 characters long.

type primary | secondary
Specifies whether the key is a primary key or a secondary key within the key chain.

Restrictions

None.

ip-router authentication create key-chain

Purpose

Create a key-chain and associate an identifier with it.

Format

ip-router authentication create key-chain <*option-list*>

Mode

Configure.

Parameters

<*option-list*>
Specifies the options you are adding. Specify one of the following:

key <*string*>
Specifies a key to be included in this key chain. The key can be up to 16 characters long.

type primary | **secondary**
Specifies whether the key is a primary key or a secondary key within the key chain.

id
Specifies an integer between 1 and 255. This option is only necessary for MD5 authentication method.

Restrictions

None.

ip-router find route

Purpose

Find the active route in the RIB which the packet will use.

Format

ip-router find route ***<ip-addr>***

Mode

Configure.

Parameters

<ip-addr>
Specifies the destination of the packet.

Restrictions

None.

ip-router global add

Purpose

Add an interface or martian. Martians are invalid addresses that are rejected by the routing software.

Format

ip-router global add interface <*name-or-IPaddr*>

ip-router global add martian <*ipAddr/mask*> | **default [host] [allow]**

Mode

Configure

Parameters

interface <*name-or-IPaddr*>
Makes an interface known to the IP router.

martian <*ipAddr/mask*> | **default [host] [allow]**
Adds a martian. Specify the following options:

<*ipAddr/mask*>	The IP address and netmask for the martian.
default	Adds default martian.
host	Specifies that this martian is a host address.
allow	Allows a subset of a range that was disallowed.

Restrictions

None.

ip-router global set

Purpose

Set various global parameters required by various protocols.

Format

ip-router global set <*option-list*>

Mode

Configure

Parameters

<*option-list*>
Specify one of the following:

autonomous-system <*num1*> **loops** <*num2*>
The autonomous system number. <*num1*> sets the as number for the router. It is only required if the router is going to run BGP. Specify a number from 1 – 65534. <*num2*> controls the number of times the as may appear in the as-path. Default is 1. It is only required if the router is going to run protocols that support as-path, such as BGP.

router-id <*hostname-or-IPaddr*>
The router ID for use by BGP and OSPF. The most preferred address is any address other than 127.0.0.1 on the loopback interface. If there are no secondary addresses on the loopback interface, then the default router ID is set to the address of the first interface which is in the up state that the SSR 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.

interface <*interface-name*> | **all preference** <*num*> **down-preference** <*num*> **passive autonomous-system** <*num*>

Specify the following:

<*interface-name*> | **all**
Specify an interface that was added using the *ip-router global add interface* command, or **all** for all interfaces.

preference <*num*>
Sets the preference for routes to this interface when it is up and functioning. Specify a number from 0 – 255. Default value is 0.

down-preference <*num*>
Sets the preference for routes to this interface when it is down. Specify a number from 0 – 255. Default value is 255.

passive
Prevents changing of route preference to this interface if it is down.

autonomous-system <*num*>
The AS that will be used to create as-path associated with the route created from the definition of this interface.

Restrictions

None.

ip-router global set trace-options

Purpose

Set various trace options.

Format

ip-router global set trace-options <*option-list*>

Mode

Configure

Parameters

<*option-list*>
Specifies the trace options you are setting. Specify one or more of the following:

startup	Trace startup events.
parse	Trace lexical analyzer and parser of gate-d config files.
ydebug	Trace lexical analyzer and parser in detail.
adv	Trace allocation and freeing of policy blocks.
symbols	Trace symbols read from kernel at startup.
iflist	Trace the reading of the kernel interface list.
all	Tun on all tracing.
general	Turn on normal and route tracing
state	Trace state machine transitions in protocols.
normal	Trace normal protocol occurrences. Abnormal occurrences are always traced.
policy	Traces the application of policy to routes being exported and imported.
task	Traces system interfaces and task processing associated with this protocol or peer.
timer	Traces timer usage by this protocol or peer
route	Traces routing table changes for routes installed by this protocol or peer.

Restrictions

None.

ip-router global set trace-state

Purpose

Enable or disable tracing.

Format

ip-router global set trace-state on | off

Mode

Configure

Parameters

on | off Specifies whether you are enabling or disabling tracing. Specify **on** to enable tracing or specify **off** to disable tracing. The default is **off**.

Restrictions

None.

ip-router global use provided_config

Purpose

Causes the SSR to use the configuration file stored in the Control Module's NVRAM.

Format

ip-router global use provided_config

Mode

Configure

Parameters

None.

Note: This command requires that you first copy the GateD configuration into the Control Module's NVRAM.

To do this, enter the following command in Enable mode:

```
ssr# copy tftp-server to gated.conf
TFTP server [10.50.89.88]? 10.50.89.88
Source filename [tmp/gated.conf]?
########
%TFTP-I-XFERRATE  Received 5910 bytes in 0.1 seconds
```

Restrictions

None.

ip-router kernel trace

Purpose

Provides trace capabilities between the Routing Information Base and the Forwarding Information Base.

Format

ip-router kernel trace <*option-list*> **detail | send | receive**

Mode

Configure

Parameters

<*option-list*>
Specifies the kernel trace options. Specify one or more of the following:

packets	Packets exchanged with the kernel.
routes	Routes exchanged with the kernel.
redirect	Redirect messages received from the kernel.
interface	Interface messages received from the kernel.
other	All other messages received from the kernel.
remnants	Routes read from the kernel when the SSR routing process starts.
request	The SSR routing process requests to Add/Delete/Change routes in the kernel forwarding table.
info	Informational messages received from the routing socket, such as TCP loss, routing lookup failure, and route resolution request.

Restrictions

None.

ip-router policy add filter

Purpose

Adds a route filter. Routes are specified by a set of filters that will match a certain set of routes by destination, or by destination and mask.

Format

ip-router policy add filter <*number-or-string*> **network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**][host-net]**

Mode

Configure

Parameters

filter <*number-or-string*>
Specifies the identifier of the route filter.

network <*IP-address*>
Specifies networks that are to be filtered. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be filtered are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
Specifies that the mask of the routes to be filtered must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

refines
Specifies that the mask of the routes to be filtered must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

host-net
This option qualifies that the specified network is a host. To match, the address must exactly match the specified and the network mask must be a host mask (i.e. all ones). This is equivalent to a network specification of host/255.255.255.255 along with the exact option.

Restrictions

None.

ip-router policy add optional-attributes-list

Purpose

Expands a previously created optional-attributes-list.

Format

ip-router policy add optional-attributes-list <*option-list*>

Mode

Configure

Parameters

<*option-list*>
Specifies the options. Specify one or more of the following:

optional-attributes-list <*number-or-string*>
Specifies the identifier for the optional attributes list you are expanding.

community-id <*number*>
Specifies a community identifier portion of a community split. This is combined with the autonomous system value entered to create a value for the community attribute.

autonomous-system <*number*>
Specifies the autonomous system portion of a community split. This would be combined with the community id value entered to create a value for the community attribute. Specify a number from 1 – 65534.

no-export
Specifies that all routes received with this attribute value ***will not*** be advertised outside a BGP confederation boundary.

well-known-community
Specifies one of the well-known communities.

no-advertise
Specifies that all routes received with this attribute value ***will not*** be advertised to other BGP peers.

no-export-subconfed
Specifies that all routes received with this attribute value ***will not*** be advertised to

external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

reserved-community <*number*>
Specifies one of the reserved communities which is not well-known. A reserved community is one which is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

Restrictions

None.

ip-router policy aggr-gen destination

Purpose

Creates an aggregate or generate route.

Format

ip-router policy aggr-gen destination <*number-or-string*> **[source** <*number-or-string*> **[filter** <*number-or-string*>**| [network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**]** **[preference** <*number*>**| restrict]]]]**

Mode

Configure

Parameters

destination <*number-or-string*>
Is the identifier of the aggregate-destination that specifies the aggregate/summarized route.

source <*number-or-string*>
Is the identifier of the aggregate-source that contributes to an aggregate route.

filter <*number-or-string*>
Specifies the filter for an aggregate/generate.

network <*ipAddr/mask*>
This option specifies networks which are to be aggregated. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be aggregated are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
This option specifies that the mask of the routes to be aggregated must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

refines
This option specifies that the mask of the routes to be aggregated must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

preference <*number*>
This option specifies the preference to be assigned to the resulting aggregate route.

Restrictions

None.

ip-router policy create aggregate-export-source

Purpose

Creates a source for exporting aggregate routes into other protocols.

Format

ip-router policy create aggregate-export-source
<*number-or-string*> **[metric** <*number*> **| restrict]**

Mode

Configure

Parameters

<*number-or-string*>	Specifies the identifier of the aggregate export source.
metric <*number*>	Specifies the metric to be associated with the exported routes.
restrict	Specifies that nothing is exported from the specified source.

Restrictions

None.

ip-router policy create aggr-gen-dest

Purpose

Creates an aggregate-generation destination. An aggregate-generation destination is one of the building blocks needed to create an aggregate/generate route.

Format

ip-router policy create aggr-gen-dest <*number-or-string*>
network <*ipAddr/mask*> | **default [type aggregate** | **generation] [preference** <*number*>**][brief]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies the identifier of an aggregate-generation destination.

network <*ipAddr/mask*> | **default**
Specifies the aggregate or generated route.

type aggregate
Specifies that the destination is an aggregate.

type generation
Specifies that the destination is a generate.

preference <*num*>
Specifies the preference to be assigned to the resulting aggregate route. The default preference is 130.

brief
Used to specify that the AS path should be truncated to the longest common AS path. The default is to build an AS patch consisting of SETs and SEQUENCES of all contributing AS paths.

Restrictions

None.

ip-router policy create aggr-gen-source

Purpose

Creates a source for the routes contributing to a aggregate/generate route.

Format

ip-router policy create aggr-gen-source <*number-or-string*>
protocol all | static | direct | aggregate | rip | ospf | bgp [autonomous-system <*number*>**][aspath-regular-expression** <*string*>**][tag** <*number*>**][preference** <*number*> **| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies the identifier of an aggregate-generation source.

protocol <*string*>
Specifies the protocol of the contributing aggregate source. Specify one of the following:

- **all**
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

autonomous-system <*number*>
Restricts selection of routes to those learned from the specified autonomous system. This selection may also be carried out by using route filters to explicitly list the set of routes to be accepted. Specify a number from 1 – 65534.

aspath-regular-expression <*string*>
Restricts selection of routes to those specified by the aspath.

tag <*number*>
Restricts selection of routes to those identified by a tag.

preference <*number*>
Specifies the preference to assign to the contributing routes.

restrict
Indicates that these routes cannot contribute to the aggregate.

Restrictions

None.

ip-router policy create aspath-export-source

Purpose

Create an export source where routes to be exported are identified by the autonomous system path associated with them. This command applies only if you are using BGP.

Format

ip-router policy create aspath-export-source <*number-or-string*> <*option-list*>

Mode

Configure

Parameters

<*number-or-string*>
Specifies a name or number for the Autonomous System path export source.

<*option-list*>
Specifies the Autonomous System path source options you are setting. Specify one of the following:

protocol <*name*>
Specifies the protocol by which the routes to be exported were learned. Specify one of the following:

- all
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

aspath-regular-expression <*string*>
Specifies an aspath regular expression which should be satisfied for the route to be exported.

origin <*string*>
Specifies whether the origin of the routes to be exported was an interior gateway protocol or an exterior gateway protocol. Specify one of the following:

- **any**
- **igp**
- **egp**
- **incomplete**

metric <*num*>
Specifies metric associated with the exported routes.

restrict
Specifies that nothing is exported from the specified source.

Note: You can specify **metric** or **restrict** even if you specified **protocol**, **aspath-regular-expression**, or **origin**.

Restrictions

None.

ip-router policy create bgp-export-destination

Purpose

Create an export destination for BGP routes.

Format

ip-router policy create bgp-export-destination
<*number-or-string*> <*option-list*>

Mode

Configure

Parameters

<*number-or-string*>
Creates a BGP export destination and associates an identifier (tag) with it.

<*option-list*>
Specifies the BGP export destination options you are setting. Specify the following:

autonomous-system <*num*>
Specifies the autonomous system of the peer-group to which we would be exporting. Specify a number from 1 – 65535.

optional-attribute-list <*num-or-string*>
Specifies the identifier of the optional-attribute-list which contains the optional attributes which are to be sent along with these exported routes. This option may be used to send the BGP community attribute. Any communities specified in the optional-attributes-list are sent in addition to any received with the route or those specified with the 'set peer-group' or 'set peer-host' commands.

metric <*num*>
Specifies the metric to be associated with the BGP exported routes.

restrict
Restricts the export of BGP routes to the specified destination.

sequence-number <*num*>
Specifies the relative position of this export-destination in a list of bgp export-destinations.

Restrictions

None.

ip-router policy create bgp-export-source

Purpose

Create a source for exporting bgp routes into other protocols.

Format

ip-router policy create bgp-export-source <*number-or-string*> <*option-list*>

Mode

Configure

Parameters

<*number-or-string*>
Creates a BGP export source and associates an identifier (tag) with it.

<*option-list*>
Specifies the BGP export source options you are setting. Specify the following:

autonomous-system <*num*>
Specifies the autonomous system of the peer-group from which we would be exporting. A route filter could alternatively be used to explicitly list a set of routes to be accepted. Specify a number from 1 – 65534.

metric <*num*>
Specifies the metric to be associated with the BGP exported routes.

restrict
Restricts the export of BGP routes from the specified source.

Restrictions

None.

ip-router policy create bgp-import-source

Purpose

Create a source for importing BGP routes.

Format

ip-router policy create bgp-import-source <*number-or-string*> <*option-list*>

Mode

Configure

Parameters

<*number-or-string*>
Creates a BGP import source and associates an identifier (tag) with it.

<*option-list*>
Specifies the BGP import source options you are setting. Specify the following:

autonomous-system <*num*>
Specifies the autonomous system of the peer-group from which we would be exporting. A route filter could alternatively be used to explicitly list a set of routes to be accepted. Specify a number from 1 – 65534.

aspath-regular-expression <*string*>
Specifies the as path regular expression that must be satisfied for the route to be exported. A route filter could alternatively be used to explicitly list a set of routes to be announced.

origin <*value*>
Specifies the origin attribute. Specify one of the following:

any Specifies that the origin attribute can be any one of **igp**, **egp** and **incomplete**.

igp Specifies that the origin attribute of the imported routes is IGP.

egp Specifies that the origin attribute of the imported routes is EGP.

incomplete Specifies that the origin attribute of the imported routes is incomplete.

optional-attribute-list <*num-or-string*>
Specifies the identifier of the optional-attribute-list. This option allows the

specification of import policy based on the path attributes found in the BGP update. If multiple communities are specified in the aspath-opt option, only updates carrying all of the specified communities will be matched. If none is specified, only updates lacking the community attribute will be matched.

preference <*num*>
Specifies the preference to be associated with the BGP imported routes.

restrict
Specifies that nothing is exported from the specified source.

sequence number <*num*>
Indicates the position this bgp import source will have in a list of BGP import sources.

Restrictions

None.

ip-router policy create direct-export-source

Purpose

Creates an export source for interface routes.

Format

ip-router policy create direct-export-source <*number-or-string*> **[interface** <*name-or-IPaddr*>**][metric** <*num*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Creates a source for exporting **interface** (**direct**) routes and associates an identifier with it.

interface
This option qualifies that the direct routes should be associated with the specific interface.

metric <*num*>
Specifies the metric to be associated with the exported routes.

restrict
Restricts the export of routes from the specified source.

Restrictions

None.

ip-router policy create filter

Purpose

Creates a route filter. Routes are filtered by specifying a set of filters that will match a certain set of routes by destination, or by destination and mask.

Format

ip-router policy create filter <*number-or-string*> **network** <*ipAddr/mask*> **[exact | refines | between** <*low-high*>**][host-net]**

Mode

Configure

Parameters

filter <*number-or-string*>
Specifies the identifier of the route filter.

network <*IP-address*>
This option specifies networks which are to be filtered. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be filtered are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
This option specifies that the mask of the routes to be filtered must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

refines
This option specifies that the mask of the routes to be filtered must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

host-net
This option qualifies that the specified network is a host. To match, the address must exactly match the specified and the network mask must be a host mask (i.e. all ones). This is equivalent to a network specification of host/255.255.255.255 along with the exact option.

Restrictions

None.

ip-router policy create optional-attributes-list

Purpose

Creates an optional-attributes-list for BGP.

Format

ip-router policy create optional-attributes-list <*option-list*>

Mode

Configure

Parameters

<*option-list*>
Specifies the options you are setting. Specify the following:

<*number-or-string*>
Specifies the identifier for the attributes list.

community-id <*number*>
Specifies a community identifier portion of a community split. This is combined with the autonomous system value entered to create a value for the community attribute.

autonomous-system <*number*>
Specifies the autonomous system portion of a community split. This would be combined with the community id value entered to create a value for the community attribute. Specify a number from 1 – 65534.

no-export
Specifies that all routes received with this attribute value ***will not*** be advertised outside a BGP confederation boundary.

well-known-community
Specifies one of the well-known communities.

no-advertise
Specifies that all routes received with this attribute value ***will not*** be advertised to other BGP peers.

no-export-subconfed
Specifies that all routes received with this attribute value ***will not*** be advertised to

external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation).

reserved-community <*number*>
Specifies one of the reserved communities which is not well-known. A reserved community is one which is in one of the following ranges (0x0000000 - 0x0000FFFF) or (0xFFFF0000 - 0xFFFFFFFF).

Restrictions

None.

ip-router policy create ospf-export-destination

Purpose

Create a destination for exporting routes into OSPF.

Format

ip-router policy create ospf-export-destination
<*number-or-string*> **[tag** <*num*>**][type 1 | 2][metric** <*num*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Creates an OSPF export destination and associates an identifier with it.

tag <*num*>
Tag to be associated with exported OSPF routes.

type 1 | 2
Specifies that OSPF routes to be exported are type 1 or type 2 ASE routes. Specify 1 or 2.

metric <*num*>
Specifies the metric to be associated with the exported routes.

restrict
Restricts the export of the specified routes.

Restrictions

It is not possible to create OSPF intra- or inter-area routes by exporting routes from the routing table into OSPF. You can only export from the routing table into OSPF ASE routes.

ip-router policy create ospf-export-source

Purpose

Create a source for exporting OSPF routes into other protocols.

Format

ip-router policy create ospf-export-source
<*number-or-string*> **[type ospf | ospf-ase][metric** <*num*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Creates an OSPF export source and associates an identifier with it.

type ospf
Exported routes are OSPF routes.

type ospf-ase
Exported routes are OSPF ASE routes.

metric <*num*>
Specifies the metric to be associated with the exported routes.

restrict
Specifies that nothing is to be exported from this source.

Restrictions

None.

ip-router policy create ospf-import-source

Purpose

Create a source for importing OSPF routes.

Format

ip-router policy create ospf-import-source <*number-or-string*> **[tag** <*num*>**][preference** <*num*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Creates an OSPF import source and associates an identifier with it.

tag <*num*>
Tag to be associated with the imported routes.

preference <*num*>
Preference associated with the imported OSPF routes.

restrict
Specifies that matching **ospf-ase** routes are not imported.

Restrictions

None.

ip-router policy create rip-export-destination

Purpose

Create a destination for exporting routes into RIP.

Format

ip-router policy create rip-export-destination <*number-or-string*>
[interface <*name-or-IPaddr*> **| gateway** <*name-or-IPaddr*>**] [metric** <*num*> **| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies an identifier for the RIP export destination:

interface <*name-or-IPaddr*> **| all**
Specifies router interfaces over which to export routes. Specify **all** to export routes to all interfaces.

gateway <*name-or-IPaddr*>
Specifies the gateway that will receive the exported routes.

metric <*num*>
Specifies the metric to be associated with the exported routes. Specify a number from 1 – 16.

restrict
Restricts the export of routes to the specified destination.

Restrictions

None.

ip-router policy create rip-export-source

Purpose

Create a source for exporting RIP routes into other protocols

Format

ip-router policy create rip-export-source
<*number-or-string*> **[interface** <*name-or-IPaddr*> **| gateway** <*name-or-IPaddr*>**][metric** <*num*> **| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies an identifier for the RIP export source:

interface <*name-or-IPaddr*>
Indicates that only routes learned over specified interfaces are exported.

gateway <*name-or-IPaddr*>
Indicates that only routes learned over specified gateways are exported.

metric <*num*>
Specifies the metric to be associated with the exported routes.

restrict
Indicates that nothing is exported from the specified source.

Restrictions

None.

ip-router policy create rip-import-source

Purpose

Create a source for importing RIP routes.

Format

ip-router policy create rip-import-source <*number-or-string*>
[interface <*name-or-IPaddr*> | **gateway** <*name-or-IPaddr*>**][preference** <*num*> | **restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies an identifier for the RIP import source:

interface <*name-or-IPaddr*>
Indicates that only routes learned over specified interfaces are imported.

gateway <*name-or-IPaddr*>
Indicates that only routes learned over specified gateways are imported.

preference <*num*>
Specifies the preference to be associated with the imported routes.

restrict
Indicates that nothing is imported from the specified source.

Restrictions

None.

ip-router policy create static-export-source

Purpose

Creates a source for exporting static routes into other protocols.

Format

ip-router policy create static-export-source <*number-or-string*>
[interface <*name-or-IPaddr*>**][metric** <*num*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Creates a source for exporting **static** routes and associates an identifier with it.

interface
This option qualifies that the **static** routes should be associated with the specific interface.

metric <*num*>
Specifies the metric to be associated with the exported routes.

restrict
Restricts the export of routes from the specified source.

Restrictions

None.

ip-router policy create tag-export-source

Purpose

Create an export source where routes to be exported are identified by the tag associated with them.

Format

ip-router policy create tag-export-source <*number-or-string*>
protocol all | static | direct | aggregate | rip | ospf | bgp
[tag <*number*>**][metric** <*number*>**| restrict]**

Mode

Configure

Parameters

<*number-or-string*>
Specifies the identifier of an tag-export source.

protocol <*string*>
Specifies the protocol of the contributing source. Specify one of the following:

- **all**
- **static**
- **direct**
- **aggregate**
- **rip**
- **ospf**
- **bgp**

tag <*number*>
Restricts selection of routes to those identified by a tag.

metric <*number*>
Specifies the metric to assign to the exported routes.

restrict
Indicates that the matching routes are not exported.

Restrictions

None.

ip-router policy export destination

Purpose

Creates an export policy from the various building blocks.

Format

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**]]]]

Mode

Configure

Parameters

<*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.

<*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*>.

<*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*>.

network <*ipAddr/mask*>
Specifies networks which are to be exported. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be exported are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
This option specifies that the mask of the routes to be exported must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

refines
This option specifies that the mask of the routes to be exported must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

restrict
Specifies that routes matching the filter are not to be exported.

metric <*number*>
Specifies the metric to be associated with the routes that match the specified filter.

Restrictions

None.

ip-router policy import source

Purpose

Creates an import policy.

Format

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

Mode

Configure

Parameters

<*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.

<*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*>.

network <*ipAddr/mask*>
Specifies networks which are to be imported. Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be imported are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
This option specifies that the mask of the routes to be imported must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network

refines
This option specifies that the mask of the routes to be imported must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

restrict
Specifies that routes matching the filter are not to be imported.

preference <*number*>
Specifies the preference with which the imported routes that match the specified filter should be installed.

Restrictions

None.

ip-router policy redistribute

Purpose

Creates a simple route redistribution policy

Format

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*>**] [tag] [ase-type]**

Mode

Configure

Parameters

from-proto <*protocol*>
Specifies the protocol of the source routes. The values for the from-proto parameter are **rip**, **ospf**, **bgp**, **direct**, **static**, **aggregate**, or **ospf-ase**.

to-proto <*protocol*>
Specifies the destination protocol where the routes are to be exported. The values for the to-proto parameter are **rip**, **ospf**, or **bgp**.

network <*ipAddr/mask*>
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.

Matching usually requires both an address and a mask, although the mask can be implied. If no additional options qualifying the networks to be redistributed are specified, then any destination that falls in the range implied by this network-specification 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. If you specify the **exact**, **refines**, or **between** parameters, the mask of the destination is also considered.

exact
This option specifies that the mask of the routes to be redistributed must match the supplied mask exactly. This is used to match a network, but not subnets or hosts of that network.

refines
This option specifies that the mask of the routes to be redistributed must be more specific (i.e. longer) than the supplied mask. This is used to match subnets.

between <*low-high*>
Specifies that the mask of the destination must be as or more specific (i.e., as long as 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 parameter).

restrict
Specifies that routes matching the filter are not to be redistributed.

metric
Indicates the metric to be associated with the redistributed routes.

tag
Tag to be associated with the exported OSPF routes.

ase-type
Routes exported from the GateD routing table into OSPF default to becoming type 1 ASEs. This default may be explicitly overridden here. Thus, this option should be used to specify if the routes are to be exported as OSPF Type 1 or Type 2 ASE routes.

Note: Each 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.

Restrictions

None.

ip-router show configuration file

Purpose

Display the active or startup configuration file in GateD format.

Format

ip-router show configuration-file active | permanent

Mode

Enable

Parameters

active	Shows the active GateD configuration file in RAM; this is the default.
permanent	Shows the permanent GateD configuration file in NVRAM, if available.

Restrictions

None.

ip-router show rib

Purpose

Display routing information base.

Format

ip-router show rib [detail]

Mode

Enable

Description

The **ip-router show rib** command shows the route-manager's routing information base (RIB). For any given network, the routing daemon could have multiple routes. The active route to any network is shown with a plus (+) sign next to it. The last active route is shown with a minus (-) next to it. If a route has been the last active route and is also the current active route, then it is shown with a asterisk (*) sign next to it. The legend is as follows:

- "+" Active Route
- "-" Last Active
- "*" Both

If the detail option is used, then additional information is displayed about these routes. The announcements bits for the active route are shown which shows the protocol into which this route is advertised.

Parameters

detail Allows you to view additional information about the routes in the RIB.

Restrictions

None.

Examples:

A sample output of the **ip-router show rib** command is shown below:

```
ssr# ip-router show rib
Routing Tables:
Generate Default: no
Destinations: 63776    Routes: 63776
Holddown: 0    Delete: 53811    Hidden: 1
Codes: Network - Destination Network Address
       S - Status + = Best Route  - = Last Active  * = Both
       Src - Source of the route :
       Ag - Aggregate  B - BGP derived  C - Connected
       R - RIP derived  St - Static  O - OSPF derived
       OE - OSPF ASE derived  D - Default
       Next hop - Gateway for the route ; Next hops in use: 4
       Netif - Next hop interface
       Prf1 - Preference of the route  Prf2 - Second Preference of the route
       Metrc1 - Metric1 of the route  Metrc2 - Metric2 of the route
       Age - Age of the route
Network/Mask       S Src Next hop             Netif Prf1 Metrc1 Metrc2       Age
------------       - --- --------             ----- ---- ------ ------       ---
3/8                *   B 134.141.178.33        mls0  170                 70:34:28
4/8                *   B 134.141.178.33        mls0  170                 70:34:28
4.17.106/24        *   B 134.141.178.33        mls0  170                 70:34:28
4.17.115/24        *   B 134.141.178.33        mls0  170                 70:34:28
4.24.148.128/25    *   B 134.141.178.33        mls0  170                 70:34:28
6/8                *   B 134.141.178.33        mls0  170                 70:34:28
6.80.137/24        *   B 134.141.178.33        mls0  170                 70:34:28
9.2/16             *   B 134.141.178.33        mls0  170                 70:34:28
9.20/17            *   B 134.141.178.33        mls0  170                 70:34:28
10.50/16           *   C 10.50.90.1              en    0      0      0 113:31:09
10.60.90/24        *   C 10.60.90.1            mls2    0      0      0 113:31:09
12/8               *   B 134.141.178.33        mls0  170                 70:34:28
12.1.248/24        *   B 134.141.178.33        mls0  170                 70:34:28
12.2.19/25         *   B 134.141.178.33        mls0  170                 12:47:48
12.2.76/24         *   B 134.141.178.33        mls0  170                 31:03:36
12.2.97/24         *   B 134.141.178.33        mls0  170                  1:41:30
12.2.109/24        *   B 134.141.178.33        mls0  170                 87:55:47
12.2.169/24        *   B 134.141.178.33        mls0  170                113:31:01
12.3.63/24         *   B 134.141.178.33        mls0  170                 70:34:28
12.4.5/24          *   B 134.141.178.33        mls0  170                 70:34:28
12.4.126/24        *   B 134.141.178.33        mls0  170                 70:34:28
12.4.164/24        *   B 134.141.178.33        mls0  170                 70:34:28
12.4.175/24        *   B 134.141.178.33        mls0  170                 95:47:57
12.4.196/22        *   B 134.141.178.33        mls0  170                 70:34:28
12.5.48/21         *   B 134.141.178.33        mls0  170                 70:34:28
12.5.164/24        *   B 134.141.178.33        mls0  170                113:31:01
12.5.252/23        *   B 134.141.178.33        mls0  170                 70:34:28
12.6.42/23         *   B 134.141.178.33        mls0  170                 70:34:28
12.6.97/24         *   B 134.141.178.33        mls0  170                 70:34:28
```

ITo see a specific route, use the **ip-router show route** command.

ip-router show route

Purpose

Displays the state of GateD.

Format

ip-router show route [ip-addr-mask | default] [detail]

Mode

Enable

Description

This command shows a specific route in the route-manager's routing information base (RIB). For any given network, the routing daemon could have multiple routes. The active route to any network is shown with a plus (+) sign next to it. The last active route is shown with a minus (-) next to it. If a route has been the last active route and is also the current active route, then it is shown with a asterisk (*) sign next to it. The legend is as follows:

- "+" Active Route
- "-" Last Active
- "*" Both

If the detail option is used, then additional information is displayed about this routes. The announcements bits for the active route are shown which shows the protocol into which this route is advertised.

Parameters

<*ipAddr/mask*> | **default**
Allows you to specify a particular IP address mask for the RIB route in question, or refer to the default address mask.

detail
Allows you to view additional information about the routes in the RIB.

Restrictions

None.

Examples

A sample output of the **ip-router show route detail** command is shown below.

```
ssr# ip-router show route 10.12.1.0/255.255.255.252 detail
10.12.1          mask 255.255.255.252
entries 2    announce 1
TSI:
RIP 150.1.255.255mc  <> metric 1
RIP 222.1.1.255mc  <> metric 1
BGP_Sync_64805 dest 10.12.1/2 metric 0
BGP group type Routing AS 64805 no metrics
Instability Histories:

*Direct    Preference:   0
*NextHop: 10.12.1.2          Interface: 10.12.1.2(to-c4500)
State: <Int Active Retain>
Age: 5:12:10    Metric: 0    Metric2: 0    Tag: 0
Task: IF
Announcement bits(5):
2-KRT 4-RIP.0.0.0.0+520 5-RIP.0.0.0.0+520
6-BGP_Sync_64805
7-BGP_Group_64805
AS Path: IGP (Id 1)

OSPF    Preference: -10
*NextHop: 10.12.1.1          Interface: 10.12.1.2(to-c4500)
State: <NotInstall NoAdvise Int Hidden Gateway>
Local AS: 64805
Age: 1:20:05   Metric: 1    Metric2: -1    Tag: 0
Task: OSPF
AS Path: (64805) IGP (Id 9551)
Cost: 1    Area: 0.0.0.0    Type: Net    AdvRouter:
172.23.1.14
```

In this case there two routes to network 10.12.1.0/255.255.255.252 One of them is a direct route and other route is learned through OSPF. The direct route has a better preference (lower preference is considered better preference), and is thus the active route. The direct route has been installed since 5 hours, 12 minutes and 10 seconds. This direct route is being announced to the Forwarding Information Base (FIB) which is indicated by KRT, over two RIP interfaces (which is indicated by 4-RIP.0.0.0.0+520, 5-RIP.0.0.0.0+520) and also to the BGP internal peer-group for autonomous system 64805.

To see all the routes in the RIB, use the **ip-router show rib** command.

ip-router show state

Purpose

Displays the state of GateD.

Format

ip-router show state [all] [memory] [timers] [to-file] [to-terminal] [task <*string*> | all | gii | icmp | inet | interface | krt | route]

Mode

Enable

Parameters

all	Shows all ouput.
memory	Shows memory allocations.
timers	Shows various GateD timers.
to-file	Saves the routing-process state in the `gated.dmp` file.
to-terminal	Displays the routing-process state on the console.
task	Shows task-specific information. The default is to show information for all tasks. You can specify a task using the **following options:**

<*string*>	Displays information for the task specified.
all	Shows information for all tasks.
gii	Shows GII information.
icmp	Shows information for the ICMP task.
inet	Shows information for the INET task.
interface	Shows information for the Interface task.
krt	Shows information for the KRT task.

route Shows information for the route task.

Restrictions

None.

Chapter 23

ip-redundancy Commands

The **ip-redundancy** commands let you display and configure the Virtual Router Redundancy Protocol (VRRP) on the SSR. VRRP is defined in RFC 2338.

Command Summary

Table 17 lists the **ip-redundancy** commands. The sections following the table describe the command syntax.

Table 17. ip-redundancy commands

ip-redundancy associate vrrp <*vrid*> **interface** <*interface*> **id** <*vrid*>
ip-redundancy clear vrrp-stats interface <*interface*> **id** <*vrid*>
ip-redundancy create vrrp <*vrid*> **interface** <*interface*>
ip-redundancy set vrrp <*vrid*> **interface** <*interface*> <*option*>
ip-redundancy show vrrp interface <*interface*> **id** <*vrid*>
ip-redundancy start vrrp <*vrid*> **interface** <*interface*>
ip-redundancy trace vrrp <*option*>

ip-redundancy associate

Purpose

Associates an IP address with a virtual router.

Format

ip-redundancy associate vrrp <*vrid*> **interface** <*interface*> **address** <*ipaddr/mask*>

Mode

Configure

Description

The **ip-redundancy associate** command adds an IP address to the list of IP addresses associated with a virtual router.

Parameters

<*vrid*>	Is the identifier of a virtual router. Specify a number between 1-255
<*interface*>	Is the name of the interface where the virtual router resides.
<*ipaddr/mask*>	Is the IP address and subnet mask to be associated with the virtual router.

Restrictions

None

Example

To add IP address/mask 1.2.3.4/16 to the list of IP addresses associated with virtual router 1 on interface int1:

```
ssr(config)# ip-redundancy associate vrrp 1 interface int1 address 1.2.3.4/16
```

ip-redundancy clear vrrp-stats

Purpose

Clears statistics gathered for VRRP.

Format

ip-redundancy clear vrrp-stats interface <*interface*> **[id** <*vrid*>**]**

Mode

Enable

Description

The **ip-redundancy clear vrrp-stats** command is used in conjunction with the **ip-redundancy show vrrp** command, which displays information about the virtual routers associated with an interface. When you specify the **verbose** option with the **ip-redundancy show vrrp** command, additional statistics are shown, including the number of times a Backup router became the Master, the number of VRRP advertisements received, and counts of VRRP packets that contain errors. When you run the **ip-redundancy clear vrrp-stats** command, these statistics are reset to zero.

Parameters

<*interface*> Causes VRRP statistics to be cleared for all virtual routers on the specified interface.

<*vrid*> Causes VRRP statistics to be cleared for the virtual router with the specified VRID. Enter a number between 1-255.

Restrictions

None.

Example

To clear statistics for virtual router 1 on interface int1:

```
ssr# ip-redundancy clear vrrp-stats interface int1 id 1
```

ip-redundancy create

Purpose

Creates a virtual router.

Format

ip-redundancy create vrrp <*vrid*> **interface** <*interface*>

Mode

Configure

Description

The **ip-redundancy create** command creates a virtual router on a specified interface.

Parameters

<*vrid*> Is the identifier of the virtual router to create. Specify a number between 1-255.

<*interface*> Is the interface on which to create the virtual router.

Restrictions

None.

Example

To create a virtual router with an identifier (VRID) of 1 on interface int1:

```
ssr(config)# ip-redundancy create vrrp 1 interface int1
```

ip-redundancy set

Purpose

Sets parameters for a virtual router.

Format

ip-redundancy set vrrp <*vrid*> **interface** <*interface*> **priority** <*number*> | **adv-interval** <*number*> | **preempt-mode enabled** | **disabled** | **auth-type none** | **text auth-key** <*key*>

Mode

Configure

Description

The **ip-redundancy set** command lets you specify parameters for a virtual router, including backup priority, advertisement interval, whether the router can preempt a Master router that has a lower priority, and the type of authentication used.

Parameters

<*vrid*>	Is the identifier of a virtual router. Specify a number between 1-255.
<*interface*>	Is the name of the interface where the virtual router resides.
priority <*number*>	Specifies the backup priority to be used by this virtual router. This number must be between 1-254. The default is 100. The priority number applies only if the virtual router is not the IP address owner. The priority of the IP address owner is always 255 and cannot be changed.
adv-interval <*number*>	Is the interval between VRRP advertisements in seconds. The default is 1 second.
preempt-mode	Specifies whether the router can preempt a Master router that has a lower priority. Use one of the following keywords:

	enabled	Preempt mode is enabled. A backup router can pre-empt a lower-priority Master router.
	disabled	Pre-empt mode is disabled. A backup router cannot pre-empt a lower-priority Master router.
auth-type	Specifies the type of authentication used for VRRP exchanges between routers. Use one of the following keywords:	
	none	VRRP exchanges are not authenticated (the default).
	text	VRRP exchanges are authenticated with a clear-text password.
auth-key <*key*>	Is the clear-text password used to authenticate VRRP exchanges. If you specify the **text** keyword, you must also specify the **auth-key** parameter.	

Restrictions

None.

Examples

To specify 200 as the priority used by virtual router 1 on interface int1:

```
ssr(config)# ip-redundancy set vrrp 1 interface int1 priority 200
```

To set the advertisement interval to 3 seconds:

```
ssr(config)# ip-redundancy set vrrp 1 interface int1 adv-interval 3
```

To prevent a Backup router from taking over as Master from a Master router that has a lower priority:

```
ssr(config)# ip-redundancy set vrrp 1 interface int1 preempt-mode disabled
```

To authenticate VRRP exchanges on virtual router 1 on interface int1 with a password of ‘yago’:

```
ssr(config)# ip-redundancy set vrrp 1 interface int1 auth-type text auth-key yago
```

ip-redundancy show

Purpose

Shows information about virtual routers.

Format

ip-redundancy show vrrp interface <*interface*> **[id** <*vrid*>**] [verbose]**

Mode

Enable

Description

The **ip-redundancy show vrrp** command displays configuration information about virtual routers on an interface. You can display information for one virtual router or for all the virtual routers on an interface. If you specify the verbose option, additional statistics are shown, including the number of times a Backup router became the Master, the number of VRRP advertisements received, and counts of VRRP packets that contain errors. These statistics are gathered from the time you start the virtual router, or from the time you last ran the **ip-redundancy clear vrrp-stats** command.

Parameters

<*interface*>	Is the name of the interface where the virtual router resides. If you do not specify the <*vrid*> parameter, information about all virtual routers on the interface is displayed.
<*vrid*>	Is the identifier of a virtual router. Specify a number between 1-255.
verbose	Causes VRRP statistics to be displayed for each virtual router

Restrictions

None.

Examples

To display information about all virtual routers on interface int1:

```
ssr# ip-redundancy show vrrp interface int1

VRRP Virtual Router 100 - Interface int1
----------------------------------------
Uptime                 0 days  0 hours  0 minutes  17 seconds.
State                  Backup
Priority               100 (default value)
Virtual MAC address    00005E:000164
Advertise Interval     1 sec(s) (default value)
Preempt Mode           Enabled (default value)
Authentication         None (default value)
Primary Address        10.8.0.2
Associated Addresses   10.8.0.1
                       100.0.0.1

VRRP Virtual Router 200 - Interface int1
----------------------------------------
Uptime                 0 days  0 hours  0 minutes  17 seconds.
State                  Master
Priority               255 (default value)
Virtual MAC address    00005E:0001C8
Advertise Interval     1 sec(s) (default value)
Preempt Mode           Enabled (default value)
Authentication         None (default value)
Primary Address        10.8.0.2
Associated Addresses   10.8.0.2
```

To display VRRP statistics for virtual router 100 on interface int1:

```
ssr# ip-redundancy show vrrp 1 interface int1 verbose

VRRP Virtual Router 100 - Interface int1
----------------------------------------
Uptime                 0 days  0 hours  0 minutes  17 seconds.
State                  Backup
Priority               100 (default value)
Virtual MAC address    00005E:000164
Advertise Interval     1 sec(s) (default value)
Preempt Mode           Enabled (default value)
Authentication         None (default value)
Primary Address        10.8.0.2
Associated Addresses   10.8.0.1
                       100.0.0.1

Stats:
  Number of transitions to master state            2
  VRRP advertisements rcvd                         0
  VRRP packets sent with 0 priority                1
  VRRP packets rcvd with 0 priority                0
  VRRP packets rcvd with IP-address list mismatch  0
  VRRP packets rcvd with auth-type mismatch        0
  VRRP packets rcvd with checksum error            0
  VRRP packets rcvd with invalid version           0
  VRRP packets rcvd with invalid VR-Id             0
  VRRP packets rcvd with invalid adv-interval      0
  VRRP packets rcvd with invalid TTL               0
  VRRP packets rcvd with invalid 'type' field      0
  VRRP packets rcvd with invalid auth-type         0
  VRRP packets rcvd with invalid auth-key          0
```

ip-redundancy start vrrp

Purpose

Starts a virtual router.

Format

ip-redundancy start vrrp <*vrid*> **interface** <*interface*>

Mode

Configure

Description

The **ip-redundancy start vrrp** command starts a virtual router on the specified interface.

Parameters

<*vrid*>	Is the identifier of a virtual router. Specify a number between 1-255.
<*interface*>	Is the name of the interface where the virtual router resides.

Restrictions

None.

Example

To start virtual router 1 on interface int1:

```
ssr# ip-redundancy start vrrp 1 interface int1
```

ip-redundancy trace

Purpose

Traces VRRP events.

Format

ip-redundancy trace vrrp events | state-transitions | packet-errors

ip-redundancy trace vrrp all enabled | disabled

Mode

Configure

Description

The **ip-redundancy trace vrrp** command displays messages when certain VRRP events take place on the SSR. Use this command to display messages when a virtual router changes from one state to another (i.e., from Backup to Master), a VRRP packet error is detected, or when any VRRP event occurs.

Parameters

events	Displays a message when VRRP receives any type of event. This option is disabled by default.
state-transitions	Displays a message when a VRRP router changes from one state to another. This option is enabled by default.
packet-errors	Displays a message when a VRRP packet error is detected. This option is enabled by default.
all enabled | disabled	Enables or disables all VRRP tracing.

Restrictions

None.

Chapter 24

ipx Commands

The **ipx** commands let you add entries to the IPX SAP table for SAP servers and display the IPX forwarding database, RIP table, and SAP table.

Command Summary

Table 18 lists the **ipx** commands. The sections following the table describe the command syntax.

Table 18. ipx commands

ipx add route <*networkaddr*> <*nextroutnextnode*> <*metric*> <*ticks*>
ipx add sap <*type*> <*SrvcName*> <*node*> <*socket*> <*metric*> <*interface-network*>
ipx find rip <*address*>
ipx find sap <*type*> | **all** <*SrvcName*> | **all** <*network*> | **all** <*entrytype*>
ipx set rip buffers
ipx set ripreq buffers
ipx set sap buffers
ipx set sapgns buffers
ipx set type20 propagation
ipx show buffers
ipx show interfaces <*interface*>

Table 18. ipx commands (Continued)

ipx show rib <*destination*>
ipx show servers hops | **net** | **name** | **type**
ipx show tables routing | **rip** | **sap** | **summary**

ipx add route

Purpose

Add an IPX RIP route entry to the routing table.

Format

ipx add route <*networkaddr*> <*nextroutnextnode*> <*metric*> <*ticks*>

Mode

Configure

Description

The **ipx add route** command adds a route into the IPX RIP routing table.

Parameters

<*networkaddr*>	Destination network address.
<*nextroutnextnode*>	Next router's Network.Node address.
<*metric*>	The number of hops to this route. You can specify a number from 0 – 14.
<*ticks*>	Ticks associated with this route.

Restrictions

Route entries that you add using the **ipx add route** command override dynamically learned entries, regardless of hop count.

Example

To add an IPX route to IPX network A1B2C3F5 via router A1B2C3D4.00:E0:63:11:11:11 with a metric of 1 and a tick of 100:

```
ssr(config)# ipx add route A1B2C3F5 A1B2C3D4.00:E0:63:11:11:11 1 100
```

ipx add sap

Purpose

Add an IPX SAP entry to the routing table.

Format

ipx add sap <*type*> <*SrvcName*> <*node*> <*socket*> <*metric*> <*interface-network*>

Mode

Configure

Description

The **ipx add sap** command adds an entry for an IPX server to the IPX SAP table.

Parameters

<*type*>	The type of service. Specify the service type using its hexadecimal value.
<*SrvcName*>	Name of the IPX server. You can use any characters in the name except the following: " * . / : ; < = > ? [ ] \ |] **Note:**Lowercase characters are changed to uppercase characters.
<*node*>	*The* IPX network and node address. Specify the address in the following format: <*netaddr*>**.**<*macaddr*>. Example: a1b2c3d4.aa:bb:cc:dd:ee:ff.
<*socket*>	The socket number for this SAP entry. You can specify a Hexadecimal number from 0x0 – 0xFFFF.
<*metric*>	The number of hops to the server. You can specify a number from 1 – 14.
<*interface-network*>	The interface network associated with this SAP entry.

Restrictions

SAP entries that you add using the **ipx add sap** command override dynamically learned entries, regardless of hop count. Moreover, if a dynamic route entry that is associated with the static SAP entry ages out or deleted, the SSR does not advertise the corresponding static SAP entries for the service until it relearns the route.

ipx find rip

Purpose

Find an IPX address in the routing table.

Format

ipx find rip <*address*>

Mode

Enable

Description

The **ipx find rip** command searches for an IPX address in the routing table.

Parameter

<*address*> The IPX network address of this interface. Specify the IPX address using its hexadecimal value.

Restrictions

None.

Example

To find an IPX network in the route table:

```
ssr(config)# ipx find rip A1B2C3F5
```

ipx find sap

Purpose

Find a SAP entry in the routing table.

Format

ipx find sap <*type*> | **all** <*SrvcName*> | **all** <*network*> | **all** <*entrytype*>

Mode

Enable

Description

The **ipx find sap** command searches for a SAP entry in the routing table.

Parameters

<*type*> | **all** The types of service. Specify the service type using its hexadecimal value. Specify **all** for all types of service.

<*SrvcName*> | **all**
Name of the IPX service. You can use any characters in the name except the following: “* . / : ; < = > ? [ ] \ |

Note: Lowercase characters are changed to uppercase characters.

Specify **all** for all IPX services.

<*network*> | **all**
Network on which the service resides. Specify an IPX network address in the following format: <*netaddr.*> Example: a1b2c3d4. Specify **all** for all networks.

<*entrytype*> The types of entry you want to find. Specify one of the following:

all Finds static and dynamic SAP entries.

dynamic Finds only the dynamic SAP entries.

static Finds only the static SAP entries.

Restrictions

None.

Example

To find a SAP entry in the route table:

```
ssr(config)# ipx find sap 4 FILESERVER a2b2c3d4 dynamic
```

ipx set rip buffers

Purpose

Sets the RIP socket buffer size in bytes.

Format

ipx set rip buffers <*buffer-size*>

Mode

Configure

Description

The **ipx set rip buffers** comand sets the RIP socket buffer size.

Parameter

<*buffer-size*>Specify the socket buffer size in bytes.

Restrictions

None.

ipx set ripreq buffers

Purpose

Sets the buffers for rip request packets.

Format

ipx set ripreq buffers <*buffer-size*>

Mode

Configure

Description

The **ipx set ripreq buffers** command sets the buffers for rip request packets.

Parameters

<*buffer-size*> Size of the buffer in bytes.

Restrictions

None.

ipx set sap buffers

Purpose

Sets the the SAP socket buffer size in bytes.

Format

ipx set sap buffers <*buffer-size*>

Mode

Configure

Description

The **ipx set sap buffers** comand sets the SAP socket buffer size.

Parameter

<*buffer-size*>Specify the buffer size in bytes.

Restrictions

None.

ipx set sapgns buffers

Purpose

Sets buffers for sap get nearest server packets.

Format

ipx set sapgns buffers <*buffer-size*>

Mode

Configure

Description

The **ipx set sapgns buffers** comand sets buffers for sap get nearest server packets.

Parameter

<*buffer-size*>Specify the buffer size in bytes.

Restrictions

None.

ipx set type20 propagation

Purpose

Controls the propagation of type 20 packets.

Format

ipx set type20 propagation

Mode

Configure

Description

The **ipx set type20 propagation command controls** the propagation of type 20 packets.

Parameter

None.

Restrictions

None.

ipx show buffers

Purpose

Display the RIP and SAP socket buffer sizes.

Format

ipx show buffers

Mode

Enable

Description

The **ipx show buffers** command displays the RIP and SAP socket buffer sizes.

Parameters

Restrictions

None.

ipx show interfaces

Purpose

Display the configuration of IPX interfaces.

Format

ipx show interfaces <*interface*>

Mode

Enable

Description

The **ipx show interfaces** command displays the configuration of an IPX interface. If you issue the command without specifying an interface name then the configuration of all IPX interfaces is displayed.

Parameters

<*interface*> Name of the IPX interface; for example, ssr14.

Restrictions

If you specify an interface name, the name must belong to an existing IPX interface.

Example

To display the configuration of all IPX interfaces:

```
ssr# ipx show interfaces
ssr12:
flags=9863<UP BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: _VLAN-1
        Ports: et.1.7
        IPX: A1B2C3D4.00:E0:63:11:11:11
ssr14:
flags=9863<UP BROADCAST NOTRAILERS RUNNING SIMPLEX LINK0 MULTICAST>
        VLAN: _VLAN-2
        Ports: et.1.2
        IPX: ABCD1234.00:E0:63:11:11:11
```

ipx show rib

Purpose

Show IPX RIP table output sorted by destination.

Format

ipx show rib <*destination*>

Mode

User

Description

The **ipx show rib** command displays IPX RIP table output sorted by destination.

Parameters

destination

Restrictions

None.

ipx show servers

Purpose

Show IPX server information.

Format

ipx show servers hop | net | name | type

Mode

User

Description

The **ipx show servers command displays IPX server information sorted by any or all of the optional arguments. Sorting is done based on the order of optional arguments given.**

Parameters

hop	Shows the output sorted by hop count.
net	Shows the ouput sorted by network number.
name	Shows the ouput sorted by service name.
type	Shows the ouput sorted by type.

Restrictions

None.

ipx show summary

Purpose

Show summary of the IPX RIP/SAP table.

Format

ipx show summary

Mode

User

Description

The **ipx show tables** command displays a summary of the IPX RIP/SAP table.

Parameters

None

Restrictions

None.

Chapter 25

l2-tables Commands

The **l2-tables** commands let you display various L2 tables related to MAC addresses.

Command Summary

Table 19 lists the **l2-tables** commands. The sections following the table describe the command syntax.

Table 19. l2-tables commands

l2-tables show all-flows [vlan <*VLAN-num*> **[source-mac** <*MACaddr*>**]] [undecoded]**
l2-tables show all-macs [verbose [undecoded]] **[vlan** <*VLAN-num*>**] [source] [destination] [multicast]**
l2-tables show bridge-management
l2-tables show igmp-mcast-registrations [vlan <*VLAN-num*>**]**
l2-tables show mac <*MACaddr*> **vlan** <*VLAN-num*>
l2-tables show mac-table-stats
l2-tables show port-macs <*port-list*> **| all-ports** **[[vlan** <*VLAN-num*>**] [source] [destination] [multicast] [undecoded] [no-stats] verbose]**
l2-tables show vlan-igmp-status vlan <*VLAN-num*>

l2-tables show all-flows

Purpose

Show all L2 flows (for ports in flow-bridging mode).

Format

l2-tables show all-flows [vlan <*VLAN-num*> **[source-mac** <*MACaddr*>**]] [undecoded]**

Mode

User or Enable

Description

The **l2-tables show all-flows** command shows all the L2 flows learned by the SSR. The SSR learns flows on ports that are operating in flow-bridging mode.

Parameters

vlan <*VLAN-num*>
The VLAN number associated with the flows. The VLAN number can be from 1 – 4095.

source-mac <*MACaddr*>
The source MAC address of the flows. Specify the MAC address in either of the following formats:

xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx

undecoded
Prevents the **SSR** from displaying the vendor names with the MAC addresses. Instead, the OUI of each MAC address is displayed "as is," in hexadecimal format. If you do not use this option, the **SSR** decodes the OUI and displays the vendor name.

Restrictions

None.

l2-tables show all-macs

Purpose

Show all MAC addresses currently in the L2 tables.

Format

l2-tables show all-macs [verbose [undecoded]]
[vlan <*VLAN-num*>**] [source] [destination] [multicast]**

Mode

User or Enable

Description

The **l2-tables show all-macs** command shows how many MAC addresses the SSR has in its L2 tables. You can format the displayed information based on VLAN, source MAC address, destination MAC address or multicast. If you enter the verbose option, the command also shows the individual MAC addresses.

Parameters

vlan <*VLAN-num*>	Displays only MAC addresses in the specified VLAN.
source	Displays only source addresses.
destination	Displays only destination addresses.
multicast	Displays only multicast and broadcast addresses.
verbose	Shows detailed information for each MAC address entry.
undecoded	Prevents the SSR from displaying the vendor names with the MAC addresses. Instead, the OUI of each MAC address is displayed "as is," in hexadecimal format. If you do not use this option, the SSR decodes the OUI and displays the vendor name.

Restrictions

None.

l2-tables show bridge-management

Purpose

Show information about all MAC addresses registered by the system.

Format

l2-tables show bridge-management

Mode

User or Enable

Description

The **l2-tables show bridge-management** command shows MAC addresses that have been inserted into the L2 tables for management purposes. Generally, these entries are configured so that a port forwards a frame to the Control Module if the management MAC matches the frame's destination MAC.

An example of a bridge-management MAC is Spanning Tree's bridge group address (0180C2:000000), which is be registered in the L2 tables of SSR ports on which the Spanning Tree Protocol (STP) is enabled.

Parameters

None.

Restrictions

None.

l2-tables show igmp-mcast-registrations

Purpose

Show information about multicast MAC addresses registered by IGMP.

Format

l2-tables show igmp-mcast-registrations [vlan <*VLAN-num*>**]**

Mode

User or Enable

Description

The **l2-tables show igmp-mcast-registrations** command shows the multicast MAC addresses that IGMP has registered with the L2 tables. The SSR forwards the multicast MAC addresses only to the ports that IGMP specifies.

Parameters

vlan <*VLAN-num*> Displays only the multicast MAC addresses registered for the specified VLAN.

Restrictions

None.

l2-tables show mac

Purpose

Show information about a particular MAC address.

Format

l2-tables show mac <*MACaddr*> **vlan** <*VLAN-num*>

Mode

User or Enable

Description

The **l2-tables show mac** command shows the port number on which the specified MAC address resides.

Parameters

<*MACaddr*> Is a MAC address. You can specify the address in either of the following formats:

xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx

vlan <*VLAN-num*>Displays the MAC address for this VLAN.

Restrictions

None.

l2-tables show mac-table-stats

Purpose

Show statistics for the MAC addresses in the MAC address tables.

Format

l2-tables show mac-table-stats

Mode

User or Enable

Description

The **l2-tables show mac-table-stats** command shows statistics for the master MAC address table in the Control Module and the MAC address tables on the individual ports.

Parameters

None.

Restrictions

None.

l2-tables show port-macs

Purpose

Show information about MACs residing in a port's L2 table.

Format

l2-tables show port-macs <*port-list*> | **all-ports**
[[vlan <*VLAN-num*>**] [source] [destination] [multicast] [undecoded] [no-stats] verbose]**

Mode

User or Enable

Description

The **l2-tables show port-macs** command shows the information about the learned MAC addresses in individual L2 MAC address tables. Each port has its own MAC address table. The information includes the number of source MAc addresses and the number of destination MAC addresses in the table. If you enter the **verbose** option, the MAC addresses also are displayed.

Parameters

port <*port-list*> | **all-ports**
Specifies the port(s) for which you want to display MAC address information. You can specify a single port or a comma-separated list of ports. If you use the **all-ports** keyword, MAC address information is displayed for all ports.

vlan <*VLAN-num*>
Specifies the type of MAC address for which you want to show statistics.

source
Displays statistics for only source addresses.

destination
Displays statistics for only destination addresses.

multicast
Displays statistics for only multicast and broadcast addresses.

undecoded
Displays the MAC addresses in hexadecimal format rather than undecoded format. Undecoded format dos not show the vendor name in place of the first three hexadecimal digits (example: Cabletron:33:44:55). The default is undecoded (example: 00:11:22:33:44:55).

no-stats
Lists the MAC addresses without displaying any statistics.

verbose
Shows detailed statistics for each MAC address entry.

Restrictions

None.

l2-tables show vlan-igmp-status

Purpose

Show whether IGMP is on or off on a VLAN.

Format

l2-tables show vlan-igmp-status vlan <*VLAN-num*>

Mode

Enable

Description

The **l2-tables show vlan-igmp-status** command shows the multicast MAC addresses that IGMP has registered with the L2 tables. This command also shows the ports to which the multicast MAC addresses are forwarded.

Note: For IGMP forwarding to occur for a multicast MAC address, IGMP must be enabled on the VLAN with which the MAC address is associated.

Parameters

vlan <*VLAN-num*> The VLAN number. The VLAN number can be from 1 – 4095.

Restrictions

None.

Chapter 26

lfap Commands

The **lfap** commands let you configure the LFAP client on the SSR and manage the Layer-3 IP accounting information that is delivered by TCP to an external server.

Command Summary

Table 20 lists the **lfap** commands. The sections following the table describe the command syntax.

Table 20. lfap commands

lfap set batch-interval <*number*>
lfap set batch-size <*number*>
lfap set lost-contact-interval <*number*>
lfap set poll-interval <*number*>
lfap set send-queue-max-size <*number*>
lfap set server <*IP address(es)*>
lfap set server-retry-interval <*number*>
lfap show all
lfap show configuration
lfap show servers
lfap show statistics
lfap show status
lfap start

lfap set batch-interval

Purpose

Defines the number of seconds between subsequent transmissions of flow creation and deletion information to a FAS.

Format

lfap set batch-interval<*number*>

Mode

Configure

Description

The **lfap set batch-interval** command defines the number of seconds between flow creation and deletion transmissions to a FAS.

Parameter

<*number*> The number of seconds (from 1 to 2,000, inclusive) between transmission of flow creation and deletion information (the interval). The default value is 1.

Restrictions

None

Example

To set the interval between flow creation and deletion transmissions to 5 seconds:

```
ssr(config)# lfap set batch-interval 5
```

lfap set batch-size

Purpose

Defines the number of flow creation and deletion records included in batch transmissions to a FAS.

Format

lfap set batch-size <*number*>

Mode

Configure

Description

The **lfap set batch-size** command defines the number of flow creation and deletion records included in information transmissions to a FAS.

Parameter

<*number*> The number of records (from 1 to 2,000, inclusive) contained in a transmission of flow creation and deletion information to a FAS. The default value is 32.

Restrictions

None

Example

To set the number of flow creation and deletion records contained in a batch transmission to 256:

```
ssr(config)# lfap set batch-size 256
```

lfap set lost-contact-interval

Purpose

Defines the period of time (in seconds) before the LFAP client realizes it has lost contact with a FAS.

Format

lfap set lost-contact-interval <*number*>

Mode

Configure

Description

The lfap set lost-contact-interval command allows you to define the amount of time (in seconds) the LFAP client will wait before realizing it has lost contact with a FAS and declare the connection lost.

Parameter

<*number*> The number of seconds (from 10 to 2,000, inclusive) the LFAP client waits before realizing that it has lost contact with a FAS. The default value is 60.

Restrictions

None

Example

To set the amount of time the LFAP client waits before realizing that it has lost contact with a FAS to 30 seconds:

```
ssr(config)# lfap set lost-contact-interval 30
```

lfap set poll-interval

Purpose

Sets the interval (in minutes) between transmissions of accounting information to the FAS server.

Format

lfap set poll-interval <*number*>

Mode

Configure

Description

The **lfap set poll-interval** command allows you to set the time period (in minutes) between subsequent transmissions of accounting data to the FAS server.

Parameters

<*number*> Defines the number of minutes (from 1 to 1,440, inclusive) between transmissions of accounting data to the FAS server. The default value is 15.

Restrictions

None

Example

To set the number of minutes between accounting data transmissions to the FAS server to 15 minutes:

```
ssr(config)# lfap set poll-interval 60
```

lfap set send-queue-max-size

Purpose

Sets the maximum number of LFAP messages that the send queue can hold before messages are dropped.

Format

lfap set send-queue-max-size <*number*>

Mode

Configure

Description

The **lfap set send-queue-max-size** command allows you to set the maximum number of LFAP messages that the send queue can hold before messages are dropped.

Parameters

<*number*> The maximum number of messages (from 100 to 2,000,000, inclusive) that the send queue can hold before messages are dropped. The default is 50,000.

Restrictions

An average LFAP message is approximately 100 bytes. You must consider the amount of memory available before you set a high number for the maximum number of messages in the send queue.

Example

To set the maximum send queue size to 100,000 LFAP messages:

```
ssr(config)# lfap set send-queue-max-size 100000
```

lfap set server

Purpose

Sets one or more FAS IP addresses for the LFAP client to contact.

Format

lfap set server *["]<IP address> [<IP address>] [<IP address>]["]*

Mode

Configure

Description

The **lfap set server** command allows you to set up to three FAS IP servers for the LFAP client to contact.

Parameters

<IP address> Sets the IP address of the FAS servers to contact. You may specify a maximum of three IP servers in the command line, separating each IP address with a space. However, if you specify more than one IP server, you must surround the IP addresses in the command line with double-quotes. (See "Examples" below.)

Restrictions

At least one IP server must be configured before the LFAP client can be started. Also, in order to delete an address from the list of IP servers to contact, you must enter a new **lfap set server** command line. (Simply negating the previous **lfap set server** command will not appropriately counter the initial command execution.)

Examples

To set one IP server to contact:

```
ssr (config)# lfap set server 5.5.5.5
```

To set three IP servers to contact:

```
ssr (config)# lfap set server “5.5.5.5 6.6.6.6 7.7.7.7”
```

lfap set server-retry-interval

Purpose

Sets the interval (in seconds) between the LFAP client's attempts to restore contact with a lost FAS.

Format

lfap set server-retry-interval <*number*>

Mode

Configure

Description

The **lfap set server-retry-interval** command allows you to customize the amount of time (in seconds) the LFAP client should wait before attempting to restore contact with a lost FAS. After the LFAP client has attempted to contact each server, it will then wait the specified number of seconds before attempting to resume contact.

Parameters

<*number*>	The number of seconds (from 1 to 2,000, inclusive) the LFAP client will wait before attempting to re-establish contact with a lost FAS. The default value is 60 seconds.

Restrictions

None

Example

To set the number of seconds between attempts to resume contact with a lost FAS to 45:

```
ssr(config)# lfap set server-retry-interval 45
```

lfap show all

Purpose

Displays all of the pertinent LFAP client data, including status, servers, configuration, and statistics.

Format

lfap show all

Mode

Enable

Description

The **lfap show all** command allows you to analyze the current status of the LFAP client and any servers to which it is currently connected. In the output of the command execution, you will find data pertaining to the following aspects of the LFAP client:

- LFAP Client Status (including connection status)
- LFAP Client Flow Accounting Servers (FASs)
- LFAP Client Configuration, including the following:
  - poll interval
  - batch size
  - batch interval
  - lost contact interval
  - server retry interval
- LFAP Client Statistics, including the following:
  - number of servers
  - up time

- connection successes and failures, including the following:
  - messages sent/received
  - lost information
  - flows

Parameters

None

Restrictions

None

lfap show configuration

Purpose

Displays the current LFAP client configuration information.

Format

lfap show configuration

Mode

Enable

Description

The **lfap show configuration** command allows you to view the current configuration of the LFAP client. In the output of the command execution, you will find the following LFAP client configuration data:

- Poll Interval
- Batch Size
- Batch Interval
- Lost Contact Interval
- Server Retry Interval

Parameters

None

Restrictions

None

lfap show servers

Purpose

Displays a list of server IP addresses to which the LFAP client is connected, or will try to contact.

Format

lfap show servers

Mode

Enable

Description

The **lfap show servers** command allows you to view the list of IP servers to which the LFAP client is currently connected, or will attempt to contact. In the output of the command execution, you will find a list of, at most, three IP addresses of associated FASs.

Parameters

None

Restrictions

None.

lfap show statistics

Purpose

Displays all of the LFAP client statistics on a per-server basis.

Format

lfap show statistics

Mode

Enable

Description

The **lfap show statistics** command allows you to view the current statistics of the LFAP client. In the output of the command execution, you will find data pertaining to the following LFAP client statistics:

- number of servers
- up time
- connection successes and failures, including the following:
  - messages sent/received
  - lost information
  - flows

Parameters

None

Restrictions

None

lfap show status

Purpose

Displays the present status of the LFAP client.

Format

lfap show status

Mode

Enable

Description

The **lfap show status** command allows you to view the current status of the LFAP client. In the output of the command execution, you will find the following LFAP client data:

- LFAP Client Status, defined as one of the following:
  - started
  - stopped
  - failed
- Connection Status, defined as one of the following:
  - connection established
  - connection lost
  - trying to connect

Parameters

None

Restrictions

None

lfap start

Purpose

Starts the LFAP client.

Format

lfap start

Mode

Configure

Description

The **lfap start** command issues a command to the LFAP client to attempt to connect to a FAS server in the list.

Parameters

None

Restrictions

At least one IP server must be configured before this command can execute successfully.

Chapter 27

load-balance Commands

The **load-balance** commands allow you to distribute session load across a pool of servers. These commands provide a way to load balance network traffic to multiple servers.

Command Summary

Table 21 lists the **load-balance** commands. The sections following the table describe the command syntax.

Table 21. load-balance commands

load-balance add host-to-group <*ipaddr/range*> **group-name** <*group name*> **port** <*port number*> [**weight** <*weight*>]
load-balance add host-to-vip-range <*range*> **vip-range-name** <*range name*> **port** <*port number*> [**weight** <*weight*>]
load-balance allow access-to-servers client-ip <*ipaddr/range*> **group-name** <*group name*>
load-balance create group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*> **protocol tcp | udp [persistence-level tcp | ssl]**
load-balance create vip-range-name <*range name*> **vip-range** <*range*> **virtual-port** <*port number*> **protocol tcp | udp [persistence-level tcp | ssl]**
load-balance set ftp-control-port <*port number*>
load-balance set hash-variant <*value*>
load-balance set mappings-age-timer <*timer*>

Table 21. load-balance commands (Continued)

load-balance set policy-for-group <*group name*> **policy** <*policy*>
load-balance set server-status server-ip <*ipaddr/range*> **server-port** <*port number*> **group-name** <*group name*> **status up** | **down**
load-balance show hash-stats
load-balance show source-mappings client-ip <*ipaddr*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*> **destination-host-ip** <*ipaddr*>
load-balance show statistics group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*>
load-balance show virtual-hosts group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*>

load-balance add host-to-group

Purpose

Adds a server to a previously-created group of load balancing servers.

Format

load-balance add host-to-group <*ipaddr/range*> **group-name** <*group name*> **port** <*port number*> [**weight** <*weight*>]

Mode

Configure

Description

The **load-balance add host-to-group** command lets you add a server to a server group that was previously-created with the **load-balance create group-name** command.

Parameters

<*ipaddr/range*>
The IP address of the server being added to the group, in the form a.b.c.d or a range of IP addresses in the form 10.10.1.1-10.10.1.3.

<*group name*>
The name of the group of load balancing servers.

<*port number*>
The port number to be used for load balancing communications for the server being added. Specify a number between 1 and 65535.

<*weight*>
This parameter is only valid if you specify the weighted round robin policy for this group of load balancing servers. (The **load-balance set policy-for-group** command specifies the policy for distributing workload to the servers.) The weight determines how many sessions are assigned to this server during its turn in the weighted round robin selection. Specify a number between 1 and 65535. The default value is 1.

Restrictions

None.

Examples

To add a server 10.10.13.2 to the server group ‘service2’:

```
ssr(config)# load-balance add host-to-group 10.10.13.2 group-name
service2 port 80
```

To add servers 10.10.13.3, 10.10.13.4, and 10.10.13.5 to the server group ‘service2’:

```
ssr(config)# load-balance add host-to-group 10.10.13.3-10.10.13.5
group-name service2 port 80
```

The following is an example of specifying the weighted round robin policy for distributing the workload on the server group ‘service2.’ To add servers 10.10.13.3, 10.10.13.4, and 10.10.13.5 to the server group ‘service2,’ a weight must be assigned to each server in the group:

```
ssr(config)# load-balance set policy-for-group service2 policy
weighted-round-robin
ssr(config)# load-balance add host-to-group 10.10.13.3 group-name
service2 port 80 weight 10
ssr(config)# load-balance add host-to-group 10.10.13.4 group-name
service2 port 80 weight 100
ssr(config)# load-balance add host-to-group 10.10.13.5 group-name
service2 port 80 weight 1000
```

load-balance add host-to-vip-range

Purpose

Adds a range of servers to a range of virtual IP addresses that were created with the **load-balance create vip-range-name** command.

Format

load-balance add host-to-vip-range <*range*> **vip-range-name** <*range name*> **port** <*port number*> [**weight** <*weight*>]

Mode

Configure

Description

The **load-balance add host-to-vip-range** command lets you add a range of servers to a range of virtual IP addresses that were previously created with the **load-balance create vip-range-name** command. This command adds the first server address in the range to the first virtual IP address, the second server address to the second virtual IP address, and so on. Therefore, the number of servers in the specified range must *equal* the number of virtual IP addresses; if you specified 15 virtual IP addresses with the l**oad-balance create vip-range-name** command, then you must specify a range of 15 IP addresses in the **load-balance add host-to-vip-range** command.

Parameters

<*range*>
: The IP range of the servers being added to the range, in the form 10.10.1.1-10.10.1.3. The number of servers in the range must be the same as the number of virtual IP addresses that were previously-created.

<*range name*>
: The name of the range of load balancing servers.

<*port number*>
: The port number to be used for load balancing communications for the server being added. Specify a number between 1 and 65535.

<weight>

This parameter is only valid if you specify the weighted round robin policy for this group of load balancing servers. (The **load-balance set policy-for-group** command specifies the policy for distributing workload to the servers.) The weight determines how many sessions are assigned to this server during its turn in the weighted round robin selection. Specify a number between 1 and 65535. The default value is 1.

Restrictions

None.

Examples

The following command creates the server groups 'service1' through 'service15' with virtual IP addresses 207.135.89.1 through 207.135.89.15:

```
ssr(config)# load-balance create vip-range-name service vip-range
207.135.89.1-207.135.89.15 virtual-port 80 protocol tcp
```

To add servers 10.10.13.1-10.10.13.15 to the server groups 'service1' through 'service15':

```
ssr(config)# load-balance add host-to-vip-range 10.10.13.1-10.10.13.15
vip-range-name service port 80
```

load-balance allow access-to-servers

Purpose

Allows specified hosts to access the load balancing servers without address translation.

Format

load-balance allow access-to-servers client-ip <*ipaddr/range*> **group-name** <*group name*>

Mode

Configure

Description

Load balancing causes both source and destination addresses to be translated on the SSR. It may be undesirable in some cases for a source address to be translated; for example, when data is to be updated on each individual server. The **load-balance allow access-to-servers** command lets you specify the hosts which are allowed to access a group of load balancing servers without address translation.

Note that a host that is allowed to access a group of load balancing servers without address translation *cannot* use the virtual IP address and port to access servers in the group.

Parameters

<*ipaddr/range*>
: The IP address of the host that is to be granted direct access, in the form a.b.c.d or a range of IP addresses in the form 10.10.1.1-10.10.1.3.

<*group name*>
: The name of the group of load balancing servers.

Restrictions

None.

Examples

To allow the host 10.23.4.8 to directly access the server group ‘service2’:

```
ssr(config)# load-balance allow access-to-servers client-ip 10.23.4.8
group-name service2
```

load-balance create group-name

Purpose

Creates a server group for load balancing.

Format

load-balance create group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*> **protocol tcp | udp [persistence-level tcp | ssl]**

Mode

Configure

Description

The **load-balance create group-name** command lets you create a load balancing server group and specify a unique "virtual" IP address and port number that is used by a client to access any server in the group. You must also specify the protocol (for example, TCP for HTTP and FTP sessions) to be used by the load balancing servers. After you create the group with this command, use the **load-balance add host** command to add specific server systems to the group.

Note: If you want to create many groups, each with a virtual IP address, use the **load-balance create vip-range-name** command.

Parameters

group-name <*group name*>

The name of this group of load balancing servers.

virtual-ip <*ipaddr*>

The address in the form a.b.c.d that will be used as the IP address for this group.

virtual-port <*port number*>

The port number to be used for this group. Specify a number between 1 and 65535.

Note: You cannot specify port number 20, as it is the FTP data port. If you create a group on the FTP control port for FTP, an implicit group will be created on port number 20.

protocol tcp | udp

The protocol used by this group of load balancing servers.

persistence-level tcp | ssl

The level of persistence to use for the bindings, either **tcp** (TCP) or **ssl** (secure socket layer). **tcp** is the default if the **persistence-level** parameter is not specified.

Restrictions

None.

Examples

To configure the server group 'service2':

```
ssr(config)# load-balance create group-name service2 virtual-ip
10.10.100.100 virtual-port 80 protocol tcp
```

load-balance create vip-range-name

Purpose

Creates a group of servers for load balancing.

Format

load-balance create vip-range-name <*range name*> **vip-range** <*range*> **virtual-port** <*port number*> **protocol tcp | udp [persistence-level]**

Mode

Configure

Description

The **load-balance create vip-range-name** command lets you specify a range of "virtual" IP addresses and a port number that is used by a client to access a server in the virtual IP address range. You must also specify the protocol (for example, TCP for HTTP and FTP sessions) to be used by the load balancing servers.

This command *implicitly* creates separate server groups for each virtual IP address in the specified range. The <*range name*> you specify becomes the base group name. Thus, the command **load-balance create vip-range-name myrange vip-range 207.135.89.1-207.135.89.15 virtual-port 80 protocol tcp** creates the groups 'myrange1' with virtual IP address 207.135.89.1, 'myrange2' with virtual IP address 207.135.89.2, etc. This command allows you to create *multiple* server groups, each with unique virtual IP addresses, whereas the **load-balance create group-name** command allows you to only create a *single* group with a *single* virtual IP address.

After you create groups with this command, you can use the **load-balance add host-to-group** command to identify specific server systems in each group. Or, you can use the **load-balance add host-to-vip-range** command to add a range of server IP addresses to each group.

Parameters

<*range name*>

The base group name for this range of load balancing servers.

vip-range <*range*>

The range of virtual IP addresses to be created.

virtual-port <*port number*>

The port number to be used for this virtual IP range. Specify a number between 1 and 65535.

Note: You cannot specify port number 20, as it is the FTP data port.

protocol tcp | udp

The protocol used by this virtual IP range.

persistence-level tcp | ssl

The level of persistence to use for the bindings, either **tcp** (TCP) or **ssl** (secure socket layer). **tcp** is the default if the **persistence-level** parameter is not specified.

Restrictions

None.

Examples

To configure the server groups 'service1' through 'service15':

```
ssr(config)# load-balance create vip-range-name service vip-range
207.135.89.1-207.135.89.15 virtual-port 80 protocol tcp
```

load-balance set ftp-control-port

Purpose

Specifies the port for FTP control.

Format

load-balance set ftp-control-port *<port number>*

Mode

Configure

Description

File Transfer Protocol (FTP) packets require special handling with load balancing, because IP address information is contained within the FTP packet data. You can use the **load-balance set ftp-control-port** command to specify the port number that is used for FTP control. The default is port 21.

Parameters

<port number>
Specifies the port number used for FTP control. Specify a value between 1 and 65535.

Restrictions

None.

Example

To set the FTP control port to 5000:

```
ssr(config)# load-balance set ftp-control-port 5000
```

load-balance set hash-variant

Purpose

Sets the hash variant for calculating the load-balancing mappings index.

Format

load-balance set hash-variant <*value*>

Mode

Configure

Description

The **load-balance set hash-variant** command sets the hash variant that is used to calculate the load-balancing mappings index. You will only need to set this variant if the **load-balance show hash-stats** command output shows extremely uneven distribution of hash table entries.

Parameters

<*value*>
Specifies the hash variant. Specify 0, 1, or 2. The default value is 0.

Restrictions

None.

Example

To set the hash variant to 1:

```
ssr(config)# load-balance set hash-variant 1
```

load-balance set mappings-age-timer

Purpose

Specifies the timeout for sessions between hosts and load-balancing servers.

Format

load-balance set mappings-age-timer <*timer*>

Mode

Configure

Description

A mapping between a host (source) and a load-balancing server (destination) times out after a period of non-use. The **load-balance set mappings-age-timer** command allows you to set the timeout for the mappings. The default is 3 minutes.

Parameters

<*timer*> The number of minutes before a source-destination mapping times out. Specify a value between 3-1440.

Restrictions

None.

Example

To set the timeout for load-balancing mappings to 720 minutes (12 hours):

```
ssr(config)# load-balance set mappings-age-timer 720
```

load-balance set policy-for-group

Purpose

Specifies the policy for distributing workload on load-balancing servers.

Format

load-balance set policy-for-group <*group name*> **policy** <*policy*>

Mode

Configure

Description

The **load-balance set policy-for-group** command allows you to specify how the SSR selects the server that will service a new session. The default policy for distributing workload among the load balancing servers is "round-robin," where the SSR selects the server on a rotating basis.

Parameters

<*group name*>
The name of this group of load balancing servers.

<*policy*>
One of the following keywords:

round-robin
The servers are selected sequentially (round-robin), without regard to the load on individual servers. This is the default policy.

weighted-round-robin
This policy is a variation of the round-robin policy. The SSR still selects servers in turn, but during its turn, each server takes on a number of session connections according to its assigned weight. For example, if 'server1' is assigned a weight of 1000 and 'server2' is assigned a weight of 10, then server1 will be assigned 1000 sessions during its turn and server2 will be assigned 10 sessions during its turn. If you specify this policy, then you should assign different weights to each server in the group with the **load-balance add host-to-group** or the **load-balance add host-to-vip-range** command.

least-loaded
The server with the fewest number of sessions bound to it is selected to service the new session.

Restrictions

None.

Example

To set the load-balancing policy for the server group ‘service2’ to ‘weighted round robin’:

```
ssr(config)# load-balance set policy-for-group service2 policy
weighted-round-robin
```

load-balance set server-status

Purpose

Sets the status of a load balancing server.

Format

load-balance set server-status server-ip <*ipaddr/range*> **server-port** <*port number*> **group-name** <*group name*> **status up | down**

Mode

Enable

Description

The **load-balance set server-status** command allows you to set the status of a load balancing server. When the status of a server is set to "down," no *new* sessions are directed to that server. Current sessions on the server are not affected. This command can be used when server content needs to be updated or to bring one or more backup servers online during peak usage times.

Parameters

server-ip <*ipaddr/range*>
IP address of the server whose status is to be set.

server-port <*port number*>
Port number of the server whose status is to be set.

group-name <*group name*>
Group name to which this server belongs.

status up | down
Sets the server status to up or down. Setting a server's status to down will cause new sessions *not* to be directed to the server.

Restrictions

None.

Example

To set the status for the server 10.10.1.2 to ‘down’:

```
ssr# load-balance set server-status server-ip 10.10.1.2 group-name
service2 status down
```

load-balance show hash-stats

Purpose

Displays load balancing hashing statistics.

Format

load-balance show hash-stats

Mode

Enable

Description

The **load-balance show hash-stats** command allows you to display load balancing hash statistics.

Parameters

None.

Restrictions

None.

Example

To display hash statistics:

```
ssr# load-balance show hash-stats
Total Mappings: 4502

Top 10 Hash Depths:

+---------+---------------+-----------------------+
|  Index  |   Hash Depth  |  Hash Depth Occurence |
+---------+---------------+-----------------------+
|1        |0              |11882                  |
|2        |1              |4226                   |
|3        |2              |138                    |
+---------+---------------+-----------------------+

Top 10 Hash Depth Occurences:

+---------+-----------------------+---------------+
|  Index  |  Hash Depth Occurence |   Hash Depth  |
+---------+-----------------------+---------------+
|1        |11882                  |0              |
|2        |4226                   |1              |
|3        |138                    |2              |
+---------+-----------------------+---------------+
```

load-balance show source-mappings

Purpose

Displays load balancing source-destination bindings.

Format

load-balance show source-mappings client-ip <*ipaddr*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*> **destination-host-ip** <*ipaddr*>

Mode

Enable

Description

The **load-balance show source-mappings** command allows you to display load balancing source-destination bindings.

Parameters

client-ip <*ipaddr*>
IP address of client whose mappings are to be shown.

virtual-ip <*ipaddr*>
Virtual IP address whose mappings are to be shown.

virtual-port <*port number*>
Virtual port number whose mappings are to be shown.

destination-host-ip <*ipaddr*>
IP address of the destination server whose mappings are to be shown.

Restrictions

None.

Example

To display source-destination bindings:

```
ssr# load-balance show source-mappings

Current Mappings:

FC: Flow Count
AC: Age Count
SPort: Source Port
VPort: Virtual Port
DPort: Destination Port

+----------------+-----+---------------+-----+----------------+-----+----+----+
| Source Address |Sport| Virtual IP |VPort| Dst. Address |DPort| FC | AC |
+----------------+-----+---------------+-----+----------------+-----+----+----+
|70.1.0.71 |1024 |50.1.1.18 |80 |52.1.1.73 |80 |2 |0 |
|70.1.0.71 |1025 |50.1.1.17 |80 |52.1.1.71 |80 |2 |0 |
|70.1.0.72 |1026 |50.1.1.17 |80 |52.1.1.72 |80 |2 |0 |
|70.1.0.72 |1027 |50.1.1.18 |80 |52.1.1.74 |80 |2 |0 |

4 source mapping(s) displayed.
```

load-balance show statistics

Purpose

Displays load balancing statistics.

Format

load-balance show statistics group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*>

Mode

Enable

Description

The **load-balance show statistics** command allows you to display load balancing statistics.

Parameters

group-name <*group name*>
Name of the group whose statistics are to be shown.

virtual-ip <*ipaddr*>
Virtual IP address whose statistics are to be shown.

virtual-port <*port number*>
Virtual port number whose statistics are to be shown.

Restrictions

None.

Example

To display load balance statistics:

```
ssr# load-balance show statistics

Load Balancing Packets Dropped:
        No Such Virtual-IP Packet drop count: 73
        TTL expired Packet drop count: 0

Load Balance Group Statistics:

        Group Name: telnet Virtual-IP: 50.1.1.17 Virtual-Port: 23
                No destination selected Packet drop count              : 0
                Memory Allocation error Packet drop count              : 0
                No forward route found Packet drop count               : 0
                Number of Packets forwarded                            : 23437
                Channel not Load Balancing compliant Packet drop count : 0
                No hosts in group Packet drop count                    : 0
                Client in Access List Packet drop count                : 2

        Group Name: http Virtual-IP: 50.1.1.17 Virtual-Port: 80
                No destination selected Packet drop count              : 2
                Memory Allocation error Packet drop count              : 0
                No forward route found Packet drop count               : 0
                Number of Packets forwarded                            : 34429
                Channel not Load Balancing compliant Packet drop count : 0
                No hosts in group Packet drop count                    : 0
                Client in Access List Packet drop count                : 1

Statistics of 2 groups shown.
```

load-balance show virtual-hosts

Purpose

Displays hosts in a load balancing group.

Format

load-balance show virtual-hosts group-name <*group name*> **virtual-ip** <*ipaddr*> **virtual-port** <*port number*>

Mode

Enable

Description

The **load-balance show virtual-hosts** command allows you to display the hosts in a load balancing group.

Parameters

group-name <*group name*>
The load balancing group that is to be shown.

virtual-ip <*ipaddr*>
IP address of the group that is to be shown.

virtual-port <*port number*>
Port number of the group that is to be shown.

Restrictions

None.

Example

To display load balance groups:

```
ssr# load-balance show virtual-hosts

Load Balanced Groups:

Flow Mode Count: 0

OS: Operational state of server
AS: Admin state of server

+----------------+----------------+------+-------------+----------+------------+
|   Group Name   |   Virtual IP   | Port | Hosts Added | Hosts Up | Next Index |
+----------------+----------------+------+-------------+----------+------------+
|telnet          |50.1.1.17       |23    |2            |2         |0           |
+----------------+----------------+------+-------------+----------+------------+
+-------+--------------+------+----------------+----+----+------------+
| Index |    Host IP   | Port |  Client Count  | OS | AS | Load Count |
+-------+--------------+------+----------------+----+----+------------+
|0      |52.1.1.73     |23    |0               |Up  |Up  |0           |
|1      |52.1.1.74     |23    |0               |Up  |Up  |0           |
+-------+--------------+------+----------------+----+----+------------+


+----------------+----------------+------+-------------+----------+------------+
|   Group Name   |   Virtual IP   | Port | Hosts Added | Hosts Up | Next Index |
+----------------+----------------+------+-------------+----------+------------+
|http            |50.1.1.17       |80    |2            |2         |0           |
+----------------+----------------+------+-------------+----------+------------+
+-------+--------------+------+----------------+----+----+------------+
| Index |    Host IP   | Port |  Client Count  | OS | AS | Load Count |
+-------+--------------+------+----------------+----+----+------------+
|0      |52.1.1.71     |80    |0               |Up  |Up  |0           |
|1      |52.1.1.72     |80    |0               |Up  |Up  |0           |
+-------+--------------+------+----------------+----+----+------------+
```

Chapter 28

logout Command

The **logout** command ends the CLI session.

Format

logout

Mode

All modes

Description

The **logout** command ends your CLI session. If you have uncommitted changes in the scratchpad, a message warns you that the changes are not saved and gives you an opportunity to cancel the logout and save the changes.

Parameters

None.

Restrictions

None.

Chapter 29

multicast Commands

The multicast dvmrp commands let you display information about IP multicast interfaces.

Command Summary

Table 22 lists the multicast commands. The sections following the table describe the command syntax.

Table 22. multicast commands

multicast show interface [<*ipAddr*>	<*hostname*>**]**
multicast show mroutes [child <*IPaddr*>**] [group** <*ipaddr*>**] [parent** <*IPaddr*>**]**

multicast show interface

Purpose

Display information about IP multicast interfaces.

Format

multicast show interface [<*ipAddr*> | <*hostname*>**]**

Mode

Enable

Description

The **multicast show interface** command displays interfaces that are running IGMP or DVMRP.

Note: This command is a superset of the **dvmrp show interface** and **igmp show interface** commands.

Parameters

<*ipAddr*> | <*hostname*> IP address or hostname of the interface.

Restrictions

None.

Examples

To display IP multicast information about interface 10.50.89.90:

```
ssr# multicast show interface 10.50.89.90
```

The following example shows a larger listing.

```
ssr# multicast show interface

Address: 172.1.1.10 Subnet: 172.1.1/24 Met: 1 Thr: 1
Name : mls15 State: Up Querier Leaf Igmp Dvmrp

Address: 207.135.89.64 Subnet: 207.135.89.0/25 Met: 1 Thr: 1
Name : company State: Up Querier Leaf Igmp Dvmrp
Groups : 224.0.1.12
224.1.127.255
224.0.1.24
224.2.127.253
224.2.127.254

Address: 10.135.89.10 Subnet: 10.135.89.0/25 Met: 1 Thr: 1
Name : test State: Up Querier Igmp Dvmrp
Peer : 10.135.89.67 Flags: 0xe Version: 3.255

Address: 190.1.0.1 Subnet: 190.1/16 Met: 1 Thr: 1
Name : rip State: Dis

Address: 207.135.122.11 Subnet: 207.135.122.8/29 Met: 1 Thr: 1
Name : mbone State: Up Igmp Dvmrp
Peer : 207.135.122.10 Flags: 0xe Version: 3.255
Groups : 224.0.1.11
224.0.1.12
224.2.127.254
239.255.255.255
224.2.127.253

Address: 10.40.1.10 Subnet: 10.40.1/24 Met: 1 Thr: 1
Name : downstream State: Up Dvmrp
Peer : 10.40.1.1 Flags: 0xf Version: 3.255

Address: 10.100.1.1 Subnet: 10.100.1/24 Met: 1 Thr: 1
Name : dan State: Dn Dvmrp
```

multicast show mroutes

Purpose

Display the IP multicast routing table.

Format

multicast show mroutes [child <*IPaddr*>**] [group** <*ipaddr*>**] [parent** <*IPaddr*>**]**

Mode

Enable

Description

The **multicast show mroutes** command displays the IP multicast routing table entry for the specified multicast group address.

This command lists all the multicast distribution trees, showing the parent interface (from where the traffic is coming), and the children distribution interfaces (to which the traffic is being forwarded). It would also show any cache information available either in hardware forwarding mechanism or in the main processor (for software based forwarding).

Note: The cache information can be timed out when not enough traffic is present, but multicast routes can still be present. Cache information is presented in number of flows (Layer 4 sessions). Multicast routes stay at least for 5 minutes, while the hardware forwarding mechanism can time out a flow faster.
Any pruning information if present is also shown.

The search can always be narrowed by looking at a particular group, and/or looking at a particular parent interface, and/or looking at a particular child interface.
Multicast routes are not the same as DVMRP routes.

Parameters

child <*ipaddr*> Address of a child interface.

group <*ipaddr*> Address of a multicast group.

parent <*ipaddr*> Address of a parent interface.

Restrictions

None.

Examples

To display the IP multicast route entry for the group 225.0.0.10:

```
ssr# multicast show mroutes group 225.0.0.10
```

Here is a fuller example of the output from this command.

```
ssr# multicast show mroutes
Network: 130.207.8/24 Group: 224.2.1.1 Age: 99s
Parent : mbone Child: test
downstream
Source : 130.207.8.82 Pkts: 383 Flows: 1

Network: 131.120.63/24 Group: 224.2.1.1 Age: 63s
Parent : mbone Pruned Child: test Pruned
downstream Pruned
Source : 131.120.63.33 Pkts: 0 Flows: 0

Network: 147.6.65.0/25 Group: 224.2.2.1 Age: 48s
Parent : mbone Pruned Child: test Pruned
downstream Pruned
Source : 147.6.65.38 Pkts: 0 Flows: 0
```

Chapter 30

mtrace Command

Purpose

Trace multicast path between a source and a receiver

Format

mtrace <*source*> **[destination** <*IPaddr*>**] [group** <*IPaddr*>**] [max-hops** <*number*>**]**

Mode

User

Description

The **mtrace** command tracks the multicast path from a source to a receiver. A trace probe is sent in a reverse path from the receiver back to the source. As the probe passes from hop to hop, it collects information such as interface address and packet counts from each router. If the **mtrace** command is executed with only the source parameter then a multicast path is calculated from the *source* to the SSR. One can examine the multicast path between two external hosts by specifying a receiver instead of using the SSR as the default receiver.

Parameters

<*source*>	IP address of the source.
destination <*IPaddr*>	Destination IP address.

group <*IPaddr*> Multicast destination group address.

max-hops <*number*> Maximum number of hops to trace (default: 0, range: 0-32)

Restrictions

None.

Examples

To display the multicast path from IP address 2.2.2.2 to the SSR:

```
ssr# mtrace 2.2.2.2
```

To display the multicast path from 1.1.1.1 to x.y.z.w for the group 239.1.1.1:

```
ssr# mtrace 1.1.1.1 destination x.y.z.w group 239.1.1.1
```

Chapter 31

nat Commands

The **nat** commands allow you to define Network Address Translation (NAT) bindings for local (inside) and global (outside) network addresses.

Command Summary

Table 23 lists the **nat** commands. The sections following the table describe the command syntax.

Table 23. nat commands

nat create dynamic local-acl-pool <*local-acl*> **global-pool** <*ip-addr/ip-addr-range/ ip-addr-list*> **[matches-interface** <*interface*>**] [enable-ip-overload]**
nat create static protocol ip | **tcp** | **udp local-ip** <*local-ip-addr/address range*> **global-ip** <*global-ip-addr/address range*> **[local-port** <*tcp/udp-local-port*> | **any] [global-port** <*tcp/udp-global-port*> | **any]**
nat flush-dynamic-binding all | **pool-specified** [**local-acl-pool** <*local-acl*>] [**global-pool** <*ip-addr/ip-addr-range*]
nat set dynamic-binding-timeout <*minutes*> | **disable**
nat set ftp-control-port <*port number*>
nat set ftp-session-timeout <*minutes*>
nat set interface <*name*> **inside** | **outside**
nat show [translations] [timeouts] [statistics]

nat create dynamic

Purpose

Defines local and global IP address pools for dynamic address binding.

Format

nat create dynamic local-acl-pool <*local-acl*> **global-pool** <*ip-addr/ip-addr-range/ip-addr-list*> **[matches-interface** <*interface*>**] [enable-ip-overload]**

Mode

Configure

Description

The **nat create dynamic** command lets you specify the local-acl pool and global IP address pool that are to be used for dynamic address binding. With dynamic address translation, IP address bindings last only until the data flow ages out or the dynamic binding is manually deleted. Global IP addresses defined for dynamic translation are reassigned whenever they become free. The local address pool for dynamic bindings are defined via an ACL profile, while the global address pool must be specified as a single IP address, an address range, an IP address and mask, or an IP list. You can also specify multiple global pools for the same local-acl pool, if you have more than one connection to the Internet on different interfaces.

Parameters

local-acl-pool <*local-acl*>

The ACL that corresponds to the local IP address pool. The ACL may contain either **permit** or **deny** keywords. Note that only the source IP address information in the ACL is used; other ACL parameters are ignored.

global-pool <*ip-addr/ip-addr-range/ip-addr-list*>

The global address pool, defined in one of the following ways:

Asingle IP address in the form a.b.c.d

An IP address range in the form 10.10.1.1-10.10.1.50

IP address and mask in the form 1.2.0.0/255.255.0.0 or 1.2.3.0/16

A list of IP addresses, separated by spaces and enclosed in quotation marks

Note: Do not specify more than 64K global addresses.

matches-interface <*interface*>
Specifies the interface to use for multiple global pools.

enable-ip-overload
Enables Port Address Translation (PAT) if no global addresses are available from the pool. This allows many local addresses to be bound to a single global address using port numbers 1024 through 4999 (port numbers are not configurable). With PAT, multiple IP addresses can map to a single IP address with multiple numbers.

Note: Protocols like ICMP do not work with the **enable-ip-overload** option. Thus, the **ping** command will not work if this option is used.

Restrictions

None.

Examples

To configure address pools for dynamic address bindings, first configure the ACL that corresponds to the local IP address pool. In the following example, the ACL 'lcl' corresponds to IP addresses from 10.1.1.1 to 10.1.1.254:

```
ssr(config)# acl lcl permit ip 10.1.1.0/24
```

Then, specify this ACL for the local IP address pool for dynamic address bindings with global addresses 136.1.1.1 to 136.1.1.254:

```
ssr(config)# nat create dynamic local-acl-pool lcl global-pool
136.1.1.0/24
```

The following examples show the use of Port Address Translation, where the global pool consists of only two specified IP addresses. In the following example, the ACL 'lcl' corresponds to IP addresses from 10.1.1.1 to 10.1.1.254:

```
ssr(config)# acl lcl permit ip 10.1.1.0/24
```

Then, specify this ACL for the local IP address pool for dynamic address bindings with global addresses 136.1.1.1 and 136.1.1.2 with Port Address Translation enabled:

```
ssr(config)# nat create dynamic local-acl-pool lcl global-pool
136.1.1.1-136.1.1.2 enable-ip-overload
```

Port numbers 1024 through 4999 can be used for global addresses 136.1.1.1 and 136.1.1.2, so you can have a maximum of about 4000 bindings per global address.

nat create static

Purpose

Defines one-to-one binding between a local address and global address.

Format

nat create static protocol ip | tcp | udp local-ip <*local-ip-addr/address range*> **global-ip** <*global-ip-addr/address range*> **[local-port** <*tcp/udp-local-port*> **| any] [global-port** <*tcp/udp-global-port*> **| any]**

Mode

Configure

Description

The **nat create static** command lets you define fixed address translation from the local network to the global network. The binding of the local to the global address does not expire until this command is negated. If the protocol used is TCP or UDP, you can also specify port address translation (PAT).

Parameters

ip | tcp | udp
Specifies either only IP address translation, IP and TCP port address translation, or IP and UDP port address translation.

local-ip <*local-ip-addr/address range*>

Either a single IP address, in the form a.b.c.d, or an address range, in the form 10.10.1.1-10.10.1.50.

global-ip <*global-ip-addr/address range*>

Either a single IP address, in the form a.b.c.d, or an address range, in the form 10.10.1.1-10.10.1.50.

local-port <*tcp/udp-local-port*> **| any**

The local TCP or UDP port number. Specify a number between 1-65535, or **any** for no port translation. This parameter is only valid if you specified **tcp** or **udp**.

Note: The number of IP addresses in the local range should be equal to the number of IP addresses in the global range.

global-port <*tcp/udp-global-port*> | **any**

The global TCP or UDP port number. Specify a number between 1-65535, or **any** for no port translation. This parameter is only valid if you specified **tcp** or udp.

Restrictions

None.

Examples

To configure a static binding of a local and a global IP address:

```
ssr(config)# nat create static protocol ip local-ip 10.1.1.13 global-ip
136.1.1.13
```

To configure a static binding of local and global IP address ranges:

```
ssr(config)# nat create static protocol ip local-ip 10.1.1.1-10.1.1.50
global-ip 136.1.1.1-136.1.1.50
```

To configure a static binding of local and global IP and UDP port addresses:

```
ssr(config)# nat create static local-ip 10.1.1.13 global-ip 136.1.1.13
local-port 18 global-port 36 protocol udp
```

nat flush-dynamic-binding

Purpose

Deletes dynamic NAT bindings.

Format

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

Mode

Enable

Description

The **nat flush-dynamic-binding** command deletes dynamic address bindings. You can delete the dynamic address bindings for specific address pools or delete all dynamic bindings.

Parameters

all
Deletes all NAT dynamic bindings.

local-acl-pool <*local-acl*>
The ACL that corresponds to the local IP address pool.

global-pool <*ip-addr/ip-addr-range*

The global address pool, defined in one of the following ways:

Asingle IP address in the form a.b.c.d

An IP address range in the form 10.10.1.1-10.10.1.50

IP address and mask in the form 1.2.0.0/255.255.0.0 or 1.2.3.0/16

Restrictions

None.

Examples

To delete dynamic address bindings for the local address pool that corresponds to the ACL 'lcl' and the global address pool that corresponds to 136.1.1.1-136.1.1.254:

```
ssr# nat flush-dynamic-binding pool-specified local-acl-pool lcl
global-pool 136.1.1.0/24
```

To delete all dynamic address bindings:

```
ssr# nat flush-dynamic-binding all
```

nat set dynamic-binding-timeout

Purpose

Sets the timeout for dynamic NAT binding.

Format

nat set dynamic-binding-timeout <*minutes*> | **disable**

Mode

Configure

Description

Dynamic address bindings time out after a period of non-use. The **nat set dynamic-binding-timeout** command lets you set the timeout for dynamic address bindings. The default is 1440 minutes (24 hours).

Parameters

<*minutes*> The number of minutes before an dynamic address binding times out. Specify a value between 3-2880.

disable Disables timeout of dynamic address bindings.

Restrictions

None

Example

To set the timeout for dynamic address bindings to 3 minutes:

```
ssr(config)# nat set dynamic-binding-timeout 3
```

To disable timeout of dynamic address bindings:

```
ssr(config)# nat set dynamic-binding-timeout disable
```

nat set ftp-control-port

Purpose

Specifies the port for FTP control.

Format

nat set ftp-control-port <*port number*>

Mode

Configure

Description

File Transfer Protocol (FTP) packets require special handling with NAT, because IP address information is contained within the FTP packet data. You can use the **nat set ftp-control-port** command to specify the port number that is used for FTP control.

The default port for FTP control is port **21**.

Parameters

<*port number*>
Specifies the port number used for FTP control. Specify a value between 1 and 65535.

Restrictions

None.

Example

To set the FTP control port to 100:

```
ssr(config)# nat set ftp-control-port 100
```

nat set ftp-session-timeout

Purpose

Specifies the timeout for the FTP session.

Format

nat set ftp-session-timeout <*minutes*>

Mode

Configure

Description

The **nat set ftp-session-timeout** command sets the timeout for the FTP session.

The default FTP session timeout is **30** minutes.

Parameters

<*minutes*> The timeout for the FTP session. Specify a value between 3-2880.

Restrictions

None.

Example

To set the FTP session timeout to 60 minutes:

```
ssr(config)# nat set ftp-session-timeout 60
```

nat set interface

Purpose

Defines an interface as inside or outside for NAT address translation.

Format

nat set interface <*name*> **inside | outside**

Mode

Configure

Description

The **nat set interface** command allows you to define an interface as inside or outside. When NAT is enabled using the **nat create static** or **nat create dynamic** command, address translation is applied only to packets that arrive on these interfaces.

Parameters

<*name*>
Is the name of the interface to which address translation will apply.

inside | outside
Specifies the interface(s) as inside or outside.

Restrictions

None.

Examples

To create the interface '10-net' and define it as an inside interface for NAT:

```
ssr(config)# interface create ip 10-net address-netmask 10.1.1.1/24
port et.2.1
ssr(config)# nat set interface 10-net inside
```

To create the interface '192-net' and define it as an outside interface for NAT:

```
ssr(config)# interface create ip 192-net address-netmask 192.50.20.1/24
port et.2.2
ssr(config)# nat set interface 192-net outside
```

nat show

Purpose

Displays NAT information.

Format

nat show [translations <*type*>**] [timeouts] [statistics]**

Mode

Enable

Description

The **nat show** command allows you to display NAT address translations, timeouts, and statistics.

Parameters

translations <*type*>
Displays NAT translations. Specify one of the following keywords:

all
Shows all translations.

type static | dynamic | overloaded-dynamic
Shows static, dynamic, or IP overloaded dynamic translations.

local-filter-in <*local-ip-addr*>
Shows translations of the specified local IP address. The IP address must be in the form a.b.c.d.

global-filter-in <*global-ip-addr*>
Shows translations of the specified global IP address. The IP address must be in the form a.b.c.d.

timeouts
Displays the current set of timeouts.

statistics
Displays NAT statistics.

Restrictions

None.

Examples

To display active NAT translations:

```
ssr# nat show translations all

Proto Local/Inside          Global/Outside IP     Type            No. of flows
----- --------------------- --------------------- --------------- ------------
TCP   15.15.15.15:1896      100.1.1.1:1026        Dyn. ovr.       2
TCP   15.15.15.15:1897      100.1.1.1:1028        Dyn. ovr. (ftp) 0
TCP   15.15.15.15:1894      100.1.1.1:1024        Dyn. ovr.       2
TCP   15.15.15.15:1895      100.1.1.1:1025        Dyn. ovr.       2
TCP   15.15.15.15:1892      100.1.1.1:1027        Dyn. ovr. (ftp) 0
IP    10.10.10.10:*         200.1.1.1:*           Dynamic         20
IP    4.4.4.4:*             202.1.1.1:*           Static          789
```

If there are many active NAT translations, you can filter the display by specifying **local-filter-in**, **global-filter-in**, or **type** parameters for the **nat show translations** command.

To display NAT timeouts:

```
ssr# nat show timeouts

All values in minutes
Flow timeout  FTP Sess. timeout  Dynamic Sess. timeout
------------  -----------------  --------------------
   2               30                    1440
```

To display NAT statistics:

```
ssr# nat show statistics

NAT is currently: active


Interface Information
---------------------
No. of Interfaces: 2
Interface: 20net  configured as nat: outside
Interface: 15net  configured as nat: inside

STATIC Binding Information
--------------------------
No. of Static Bindings: 1

DYNAMIC Binding Information
---------------------------
No. of Dynamic Bindings: None

Local Acl pool Max. globals Globals used Max. ports Ports Used Err cnt
-------------- ------------ ------------ ---------- ---------- -------
local                 1            0        3975          0        0
```

Chapter 32

negate Command

The **negate** command negates a command in the scratchpad or the active configuration.

Format

negate <*cmd-number*> **[scratchpad | active-config]**

Mode

Configure

Description

The **negate** command allows you to negate one or more commands by specifying the command number of the commands you want to negate. The command number for each command can be found using the Configure mode **show** command. You can negate commands from the active running system or non-committed commands from the scratchpad. By default, if you do not specify **active-config** or **scratchpad**, the command to negate is assumed to be in the **active-config**.

Parameters

<*cmd-number*>	The number of the command(s) you want to negate. Use the **show** command to display the command numbers.
active-config	Negate the specified command from the active running system.
scratchpad	Negate the specified non-committed command from the scratchpad.

Restrictions

The specified command number must represent a command that exists.

Examples

To negate command 23 from the active configuration:

```
ssr# negate 23
```

To negate commands 3, 5, 6 and 7 from the scratchpad:

```
ssr# negate 3 5-7 scratchpad
```

Chapter 33

no Command

The **no** command removes a configuration command from the active configuration of the running system.

Format

no <*command-to-negate*>

Mode

Configure

Description

The **no** command allows you to negate a previously executed command. Following the keyword **no**, one can specify the command to negate in its entirety or use the wildcard character (*****) to negate a group of commands. In addition to the **no** command, one can also use the **negate** command to negate a group of commands using the command number.

Parameters

<*command*> The CLI command you want to negate. You do not have to enter the entire command. You can use the wildcard character, *, to negate matching commands. For example, if you specify “no acl 100 *” then all commands starting with the words “acl 100” will be negated.

Restrictions

The command to negate must already be in the active configuration. You cannot negate a command that hasn't been entered.

Examples

To negate the specified **arp add** command, enter the following. By negating this command, the system removes the ARP entry for *nfs2* from the ARP table.

```
ssr# no arp add nfs2 macaddr 080020:13a09f exit-port et.3.1
```

To negate all commands starting with the word "acl":

```
ssr# no acl *
```

Chapter 34

ntp Commands

The ntp commands configure and display the characteristics of the NTP (Network Time Protocol) client.

Command Summary

Table 24 lists the ntp commands. The sections following the table describe the command syntax.

Table 24. ntp commands

ntp set server <*host*> **[interval** <*minutes*>**] [source** <*ipaddr*>**] [version** <*num*>**]**
ntp show all
ntp synchronize server <*host*>

ntp set server

Purpose

Specifies the NTP server against which the SSR is to synchronize its clock.

Format

ntp set server <*host*> **[interval** <*minutes*>**] [source** <*ipaddr*>**] [version** <*num*>**]**

Mode

Configure

Description

The **ntp set server** command instructs the SSR's NTP client to periodically synchronize its clock. By default, the SSR specifies an NTPv3 client that sends a synchronization packet to the server every 60 minutes. This means the SSR 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.

Parameters

server <*host*>	Specifies the hostname or the IP address of the NTP server.
interval <*minutes*>	Specifies how often (in minutes) the SSR should synchronize with the server. The default synchronization interval is 60 minutes. Valid interval is between 1 minute to 10080 minutes (7 days).
source <*ipaddr*>	Specifies the source IP address to be used by the SSR for sending the NTP packet. The IP address must belong to one of the interfaces on the SSR.
version <*num*>	Specifies the NTP version number of the packet. The default version number is 3 (NTPv3). Valid value is 1-3.

Restrictions

None.

Examples

To send NTP packets to the NTP server 10.13.1.1 with default parameters:

```
ssr(config)# ntp set server 10.13.1.1
```

To synchronize with a NTP server every 15 minutes with a specific source IP address:

```
ssr(config)# ntp set server 10.13.1.1 interval 15 source 10.15.3.3
```

ntp show all

Purpose

Display NTP information about the SSR.

Format

ntp show all

Mode

Enable

Description

The **ntp show all** command displays various NTP information about the SSR, for example, the last time a successful synchronization was made, synchronization interval, NTP version number, etc.

Parameters

None.

Restrictions

None.

Example

```
ssr# ntp show all
NTP status:
     Synchronization interval: 60 mins
     Version: NTPv3
     Last successful contact: Thu Jan 23 23:08:15 1999
```

ntp synchronize server

Purpose

Manually force the SSR to immediately synchronize with a NTP server.

Format

ntp synchronize server <*host*>

Mode

Enable

Description

The **ntp synchronize server** command forces the SSR to immediately synchronize its clock with the NTP server. Unlike the Configuration mode **ntp set server** command, this Enable mode command does not send periodic synchronization packets to the server. Instead, each time this command is executed, the SSR synchronizes itself with the server. To have the SSR synchronizes itself periodically, use the **ntp set server** command.

Parameters

<*host*>	Specifies the hostname or the IP address of the NTP server.

Restrictions

None.

Examples

To synchronize the SSR against the NTP server 10.13.1.1:

```
ssr(config)# ntp synchronize server 10.13.1.1
%NTP-I-TIMESYNC  Time synchronized to Thu Jan 23 23:11:28 1999
```

Chapter 35

ospf Commands

The ospf commands let you display and set parameters for the Open Shortest Path First (OSPF) routing protocol.

Command Summary

Table 25 lists the ospf commands. The sections following the table describe the command syntax.

Table 25. ospf commands

ospf add interface <*interfacename-or-IPaddr*> **[to-area** <*area-addr*> **| backbone]** **[type broadcast | non-broadcast]**
ospf add nbma-neighbor <*hostname-or-IPaddr*> **to-interface** <*hostname-or-IPaddr*> **[eligible]**
ospf add stub-host [to-area <*area-addr*> **| backbone] [cost** <*num*>**]**
ospf add network | summary-range
ospf add virtual-link <*number-or-string*> **[neighbor** <*IPaddr*>**]** **[transit-area** <*area-num*>**]**
ospf create area <*area-num*> **[backbone]**
ospf create-monitor destination <*hostname-or-IPaddr*>
ospf monitor <*option-list*>
ospf set area <*area-num*> **[stub] [stub-cost** <*num*>**]** **[authentication-method none | simple | md5]**
ospf set ase-defaults [preference <*num*>**] [cost** <*num*>**]** **[type** <*num*>**] [inherit-metric]**

Table 25. ospf commands (Continued)

ospf set export-interval <*num*>
ospf set export-limit <*num*>
ospf set interface <*interfacename-or-IPaddr*> **| all** **[state disable | enable] [cost** <*num*>**] [no-multicast] [retransmit-interval** <*num*>**]** **[transit-delay** <*num*>**] [priority** <*num*>**] [hello-interval** <*num*>**]** **[router-dead-interval** <*num*>**] [poll-interval** <*num*>**][key-chain** <*num-or-string*>**]**
ospf set monitor-auth-method none | simple | md5
ospf set trace-options [lsa-build] [spf] [lsa-transmit] [lsa-receive] [state] [hello] [dd] **[request] [lsu] [ack]**
ospf set virtual-link <*number-or-string*> **[state disable | enable] [cost** <*num*>**]** **[no-multicast] [retransmit-interval** <*num*>**] [transit-delay** <*num*>**] [priority** <*num*>**] [hello-interval** <*num*>**] [router-dead-interval** <*num*>**] [poll-interval** <*num*>**]**
ospf show <*option-list*>
ospf start | stop

ospf add interface

Purpose

Associates an interface with an OSPF area.

Format

ospf add interface <*interfacename-or-IPaddr*> **[to-area** <*area-addr*> **| backbone]**
[type broadcast | non-broadcast]

Mode

Configure

Parameters

<*interfacename-or-IPaddr*>
An interface name or an IP address.

to-area <*area-addr*> **| backbone**
OSPF Area with which this interface is to be associated.

type
Specifies whether the interface is broadcast or non-broadcast. Specify one of the following:

- **broadcast** (default)
- **non-broadcast**

Restrictions

None.

ospf add nbma-neighbor

Purpose

Specifies an OSPF NBMA Neighbor.

Format

ospf add nbma-neighbor <*hostname-or-IPaddr*> **to-interface** <*interfacename-or-IPaddr*> **[eligible]**

Mode

Configure

Parameters

to-interface <*interfacename-or-IPaddr*>
Adds the neighbor to the specified OSPF interface.

eligible
Specifies whether an OSPF NBMA Neighbor is eligible for becoming a designated router.

Restrictions

None.

ospf add network | summary-range

Note: Because the **OSPF add network** command is misinterpreted with commands having similar syntax from other vendors, this command will eventually be dropped from the SSR's host of CLI commands. The new command is **ospf add summary-range**. At this time, however, both are acceptable CLI commands, hence both are dealt with in this section.

Purpose

Configures summary-ranges on Area Border Routers (ABRs). This allows you to reduce the amount of routing information propagated between areas.

On the SSR, summary-ranges are created using the **ospf add summary-range** 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/networks are advertised as summary network LSAs. If you specify the **restrict** option, the summary network LSAs are not advertised. Each intra-area LSA that does not fall into any range is advertised as an OSPF Type-3 or 4 LSA.

Format

ospf add network | summary-range <*IPaddr/mask*> **[to-area** <*area-addr*>**] [restrict] [host-net]**

Mode

Configure

Parameters

<*IPaddr/mask*>
IP Address and network mask value representing the summary-range. Example: 16.122.0.0/255.255.0.0 or 16.122.0.0/16.

to-area <*area-addr*>
OSPF Area with which this summary-range is to be associated.

restrict
If the restrict option is specified for a network/summary-range, then that network is not advertised in Summary network LSAs.

host-net
Specifies that the network is an OSPF Host Network.

Restrictions

None.

Example

In the following example, two summary ranges are created:

```
ospf add summary-range 207.135.16.0/24 to-area 207.135.0.0
ospf add summary-range 207.135.17.0/24 to-area 207.135.0.0 restrict
```

Intra-area Link State Advertisements (LSAs) that fall within the range 207.135.16.0/24 are not advertised into other areas as inter-area routes. Instead, the specified range 207.135.16.0/24 is advertised as summary network LSA.

Because the summary range 207.135.17.0/24 has the restrict option associated with it, intra-area link state advertisements (LSAs) that fall within it are not advertised as summary network LSA. Using this mechanism, one can have “hidden networks” within an area, which are not advertised to other areas.

ospf add stub-host

Purpose

Adds a stub-host to an OSPF area.

Format

ospf add stub-host <*hostname-or-IPaddr*> **[to-area** <*area-addr*>**| backbone] [cost** <*num*>**]**

Mode

Configure

Parameters

to-area <*area-addr*>**| backbone**
OSPF Area to which you are adding a stub host.

cost <*num*>
The cost that should be advertised for this directly attached stub host. Specify a number from 0 – 65535.

Restrictions

None.

ospf add virtual-link

Purpose

Creates an OSPF Virtual Link.

Format

ospf add virtual-link <*number-or-string*> **[neighbor** <*IPaddr*>**] [transit-area** <*area-num*>**]**

Mode

Configure

Parameters

<*number-or-string*>
A number or character string identifying the virtual link.

neighbor <*IPaddr*>
The IP address of an OSPF virtual link neighbor.

transit-area <*area-num*>
The Area ID of the transit area.

Restrictions

None.

ospf create area

Purpose

Create an OSPF area.

Format

ospf create area <*area-num*> | **backbone**

Mode

Configure

Parameters

<*area-num*>	The Area ID. Normally, Area IDs are formatted like IP addresses: <*num*>.<*num*>.<*num*>.<*num*>.
backbone	Specifies that the Area you are adding is the backbone area.

Restrictions

None.

ospf create-monitor

Purpose

Create an OSPF monitor destination.

Format

ospf create-monitor destination <*hostname-or-IPaddr*>

Mode

Enable

Parameters

destination <*hostname-or-IPaddr*>
Specifies the destination whose OSPF activity is to be monitored.

Restrictions

None.

ospf monitor

Purpose

Monitor OSPF.

Format

ospf monitor statistics | errors | next-hop-list | interfaces | neighbors [destination <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

ospf monitor lsdb [display-retransmit-list] [destination <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

ospf monitor routes [type all | asbrs-in-area | area-border-routers | asbrs-other-areas | networks-in-area | networks-other-areas | as-routes] [destination <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

ospf monitor lsa area-id <*IPaddr*> **type router-links | network-links | summary-networks | summary-asbr | as-external ls-id** <*IPaddr*> **adv-rtr** <*IPaddr*> **[destination** <*hostname-or-IPaddr*>**] [auth-key** <*string*>**]**

ospf monitor as-external-db [display-retransmit-list destination <*IPaddr*>**] [auth-key** <*string*>**]**

Mode

Enable

Parameters

destination <*hostname-or-IPaddr*>
Monitors the specified OSPF destination. Default is the router on which the command is executed.

auth-key <*string*>
Specifies the authorization key for the OSPF destination. This option is not needed if the OSPF destination does not require a key or if an authorization was specified using the **ospf monitor create-destination** command.

statistics
Shows input/output statistics for monitor request, hello, data base description, link-state request, link-state update, and link-state ack packets. Area statistics are

provided, which describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external data base entries.

errors
Shows the various error conditions which can occur between OSPF routing neighbors and the number of occurrences for each.

next-hop-list
Shows information about all valid next hops mostly derived from the SPF calculation.

interfaces
Shows information about all interfaces configured for OSPF. Information reported includes the area, interface IP address, interface type, interface state, cost, priority, and the IP address of the Designated Router and Backup Designated Router for the network.

neighbors
Shows information about all OSPF routing neighbors. Information reported includes the area, local interface address, router ID, neighbor IP address, state, and mode.

lsdb
Displays the link-state database (except for ASEs). This table describes the routers and networks making up the AS. If the display-retransmit-list option is specified, the retransmit list of neighbors held by this lsdb structure will also be printed.

display-retransmit-list – Displays the retransmit list from the link state database.

routes
Displays the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, networks, summary networks and AS external networks currently managed via OSPF.

type all
Shows all OSPF routes.

type asbrs-in-area
Shows routes to AS boundary routers in this area.

type area-border-routers
Shows routes to area border routers for this area.

type asbrs-other-areas
Shows summary routes to AS boundary routers in other areas.

type networks-in-area
Shows routes to networks in this area.

type networks-other-areas
Shows routes to networks in other areas.

type as-routes
Shows AS routes to non-OSPF networks.

lsa
Displays the link state advertisement. Area_Id is the OSPF area for which the query is directed. Adv_Rtr is the router -id of the router which originated this link state advertisement. Type specifies the type of advertisement to request:

area-id <*IPaddr*>
Specifies the OSPF area.

type router-links
Requests router link advertisements that describe the collected states of the router interfaces. ls-id is set to the originating router's router-id.

type network-links
Requests network link advertisements that describe the set of routers attached to the network. ls-id is set to the IP interface address of the designated router for the network.

type summary-networks
Request summary-link advertisements describing routes to networks. ls-id is set to the IP address of the destination network.

type summary-asbr
Requests summary-link advertisements describing routes to AS boundary routers. ls-id is set to the AS boundary router's router-id.

type as-external
Requests AS external link state advertisements. ls-id is set to the IP address of the destination network.

ls-id <*IPaddr*>
Species the ls-id for the type of link-state advertisement requested

adv-rtr <*IPaddr*>
Requests the router ID of the originating router.

as-external-db
Display the AS external data base entries. This table reports the advertising router, forwarding address, age, length, sequence number, type, and metric for each AS external route. If the display-retransmit-list option is specified, the retransmit list of neighbors held by this lsdb structure will also be printed.

Restrictions

None.

Examples

The following are examples of **ospf monitor** commands.

```
ssr# ospf monitor statistics

IO stats
Input  Output  Type
8       0  Monitor request
1322    1314  Hello
716     721  DB Description
39     728  Link-State Req
3037    3355  Link-State Update
1317     354  Link-State Ack
ASE: 1903 checksum sum 3BB0F22

LSAs originated: 1915   received: 17
Router: 5  ASE: 1910

Area 0.0.0.0:
Neighbors: 3    Interfaces: 3
Spf: 3  Checksum sum 6CB41
DB: rtr: 5 net: 5 sumasb: 0 sumnet: 2

Routing Table:
Intra Area: 5   Inter Area: 4   ASE: 1
```

```
ssr# ospf monitor errors

Packets Received:
10: Monitor request                   1342: Hello
716: DB Description                      39: Link-State Req
3212: Link-State Update                1536: Link-State Ack

Packets Sent:
0: Monitor response                   1335: Hello
721: DB Description                      728: Link-State Req
3907: Link-State Update                   359: Link-State Ack

Errors:
0: IP: bad destination                   0: IP: bad protocol
0: IP: received my own packet            0: OSPF: bad packet type
0: OSPF: bad version                     0: OSPF: bad checksum
0: OSPF: bad area id                     0: OSPF: area mismatch
0: OSPF: bad virtual link                0: OSPF: bad authentication type
0: OSPF: bad authentication key          0: OSPF: packet too small
0: OSPF: packet size > ip length         1: OSPF: transmit error
0: OSPF: interface down                  0: OSPF: unknown neighbor
0: HELLO: netmask mismatch               0: HELLO: hello timer mismatch
0: HELLO: dead timer mismatch            0: HELLO: extern option mismatch
0: HELLO: router id confusion            0: HELLO: virtual neighbor
unknown
0: HELLO: NBMA neighbor unknown          0: DD: neighbor state low
0: DD: router id confusion               0: DD: extern option mismatch
```

```
0: DD: unknown LSA type                1: LS ACK: neighbor state low
0: LS ACK: bad ack                  1140: LS ACK: duplicate ack
0: LS ACK: Unknown LSA type            0: LS REQ: neighbor state low
0: LS REQ: empty request               0: LS REQ: bad request
8: LS UPD: neighbor state low          0: LS UPD: newer self-gen LSA
0: LS UPD: LSA checksum bad          131: LS UPD: received less recent
LSA
0: LS UPD: unknown LSA type            2: Interface: Not configed for
OSPF
0: Interface: Invalid type             0: Interface: Mcast disabled.
0: Interface: Invalid state            0: Interface: Address not found
1: No vlinks and src is non local
```

```
ssr# ospf monitor next-hop-list

Next hops:

Address          Type      Refcount  Interface
-------------------------------------------------------------
10.12.1.1        Neighbor         6  10.12.1.2       to-c4500
10.12.1.2        Direct           1  10.12.1.2       to-c4500
150.1.0.1        Direct           1  150.1.0.1       to-aval-eth5
172.23.1.5       Direct           3  172.23.1.5      to-SSR6
172.23.1.6       Neighbor         5  172.23.1.5      to-SSR6
172.23.1.21      Direct           3  172.23.1.21     to-SSR1
172.23.1.22      Neighbor        19  172.23.1.21     to-SSR1
172.23.1.25      Direct           3  172.23.1.25     lo
222.1.1.1        Direct           1  222.1.1.1       to-linux1
```

```
ssr# ospf monitor interfaces
>sent to 127.0.0.1

Source <<127.0.0.1       >>

Area: 0.0.0.0
IP Address      Type  State    Cost Pri DR             BDR
-------------------------------------------------------------------------
172.23.1.5      Bcast BackupDR 2    2   172.23.1.6      172.23.1.5
10.12.1.2       Bcast BackupDR 1    2   10.12.1.1       10.12.1.2
172.23.1.21     Bcast BackupDR 1    2   172.23.1.22     172.23.1.21
done
```

```
ssr# ospf monitor neighbors
> sent to 127.0.0.1

Source <<127.0.0.1       >>
```

```
Interface: 172.23.1.5      Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
0.0.0.6         172.23.1.6      Full       Slave  1

Interface: 10.12.1.2       Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
172.23.1.14     10.12.1.1       Full       Slave  1

Interface: 172.23.1.21     Area: 0.0.0.0
Router Id       Nbr IP Addr     State      Mode   Prio
-----------------------------------------------------
0.0.0.1         172.23.1.22     Full       Master 1
done
```

```
ssr# ospf monitor routes
> sent to 127.0.0.1

Source <<127.0.0.1       >>
AS Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------
Area 0.0.0.0:
0.0.0.6            2 0.0.0.6         172.23.1.6
172.23.1.22
0.0.0.4            0 0.0.0.4
0.0.0.1            1 0.0.0.1         172.23.1.22

Total AS Border routes: 3

Area Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------
Area 0.0.0.0:
0.0.0.3            2 0.0.0.3         172.23.1.22
0.0.0.1            1 0.0.0.1         172.23.1.22

Total Area Border Routes: 2

Summary AS Border Routes:
Router          Cost AdvRouter       NextHop(s)
-------------------------------------------------

Networks:
Destination        Area            Cost Type NextHop         AdvRouter
-------------------------------------------------------------------------

172.23.1.4/30      0.0.0.0            2 Net  172.23.1.5      0.0.0.6
10.12.1.0/30       0.0.0.0            1 Net  10.12.1.1       172.23.1.14
172.23.1.20/30     0.0.0.0            1 Net  172.23.1.21     0.0.0.1
172.23.1.25        0.0.0.0            0 Stub 172.23.1.25     0.0.0.4
172.23.1.8/30      0.0.0.0            2 Net  172.23.1.22     0.0.0.1
10.12.1.4/30       0.0.0.0            2 Net  172.23.1.22     172.23.1.14
172.23.1.14        0.0.0.0            2 Stub 10.12.1.1       172.23.1.14
```

```
172.23.1.26        0.0.0.0            3 Stub 172.23.1.6      0.0.0.6
172.23.1.22
16                 0.0.0.0            2 SNet 172.23.1.22     0.0.0.1
ASEs:
Destination        Cost E      Tag NextHop         AdvRouter
------------------------------------------------------------------------------
15.1                 1   1 c0000000 172.23.1.22     0.0.0.1
Total nets: 9
Intra Area: 5   Inter Area: 4   ASE: 1
done
```

```
ssr# ospf monitor lsdb

LS Data Base:
Area: 0.0.0.0
Type LinkState ID    AdvRouter        Age Len Sequence Metric Where
------------------------------------------------------------------
Stub 172.23.1.25     0.0.0.4          341 24  0             0 SpfTree
Stub 172.23.1.14     172.23.1.14      352 24  0             0 SpfTree
Stub 172.23.1.26     0.0.0.6          343 24  0             0 SpfTree
Rtr  0.0.0.1         0.0.0.1          309 72  800009b0      0 SpfTree
Rtr  0.0.0.3         0.0.0.3         1223 36  80000011      0 SpfTree
Rtr  0.0.0.4         0.0.0.4          341 72  80000084      0 SpfTree
Rtr  172.23.1.14     172.23.1.14       74 60  80000bf6      0 Clist
Rtr  0.0.0.6         0.0.0.6          227 60  80000a0d      0 SpfTree
Net  172.23.1.10     0.0.0.1          309 32  80000005      0 SpfTree
Net  172.23.1.22     0.0.0.1          309 32  80000003      0 SpfTree
Net  10.12.1.1       172.23.1.14       74 32  80000002      0 SpfTree
Net  10.12.1.6       172.23.1.14       74 32  8000003d      0 SpfTree
Net  172.23.1.6      0.0.0.6          227 32  80000003      0 SpfTree
SNet 16.255.255.255  0.0.0.3         1129 28  8000000c      1 Uninitialized
SNet 16.255.255.255  0.0.0.1          215 28  80000003      1 Uninitialized
done
```

```
ssr# ospf monitor as-external-db

AS External Data Base:
Destination        AdvRouter       Forward Addr     Age Len Sequence T Metric
------------------------------------------------------------------------------
130.58.225         0.0.0.4         0.0.0.0          201 36  80000001 21
130.58.174         0.0.0.4         0.0.0.0          201 36  80000001 21
130.56.235         0.0.0.4         0.0.0.0          236 36  80000001 21
130.56.184         0.0.0.4         0.0.0.0          236 36  80000001 21
130.54.245         0.0.0.4         0.0.0.0          238 36  80000001 21
130.54.194         0.0.0.4         0.0.0.0          239 36  80000001 21
130.52.255         0.0.0.4         0.0.0.0          241 36  80000001 21
130.52.204         0.0.0.4         0.0.0.0          241 36  80000001 21
130.51.9           0.0.0.4         0.0.0.0          211 36  80000001 21
130.50.214         0.0.0.4         0.0.0.0          211 36  80000001 21
130.49.19          0.0.0.4         0.0.0.0          213 36  80000001 21
130.48.224         0.0.0.4         0.0.0.0          214 36  80000001 21
```

```
130.47.29       0.0.0.4         0.0.0.0         216 36  80000001 21
130.46.234      0.0.0.4         0.0.0.0         248 36  80000001 21
130.45.39       0.0.0.4         0.0.0.0         251 36  80000001 21
130.44.244      0.0.0.4         0.0.0.0         251 36  80000001 21
130.43.49       0.0.0.4         0.0.0.0         253 36  80000001 21
130.42.254      0.0.0.4         0.0.0.0         221 36  80000001 21
130.41.59       0.0.0.4         0.0.0.0         256 36  80000001 21
130.41.8        0.0.0.4         0.0.0.0         256 36  80000001 21
130.39.69       0.0.0.4         0.0.0.0         258 36  80000001 21
130.39.18       0.0.0.4         0.0.0.0         258 36  80000001 21
130.37.79       0.0.0.4         0.0.0.0         261 36  80000001 21
130.37.28       0.0.0.4         0.0.0.0         261 36  80000001 21
130.35.89       0.0.0.4         0.0.0.0         263 36  80000001 21
130.35.38       0.0.0.4         0.0.0.0         263 36  80000001 21
130.33.99       0.0.0.4         0.0.0.0         267 36  80000001 21
130.33.48       0.0.0.4         0.0.0.0         267 36  80000001 21
130.31.109      0.0.0.4         0.0.0.0         272 36  80000001 21
130.31.58       0.0.0.4         0.0.0.0         272 36  80000001 21
130.29.119      0.0.0.4         0.0.0.0         277 36  80000001 21
130.29.68       0.0.0.4         0.0.0.0         277 36  80000001 21
130.27.129      0.0.0.4         0.0.0.0         282 36  80000001 21
130.27.78       0.0.0.4         0.0.0.0         282 36  80000001 21
130.25.139      0.0.0.4         0.0.0.0         287 36  80000001 21
130.25.88       0.0.0.4         0.0.0.0         287 36  80000001 21
130.23.149      0.0.0.4         0.0.0.0         292 36  80000001 21
130.23.98       0.0.0.4         0.0.0.0         292 36  80000001 21
130.21.159      0.0.0.4         0.0.0.0         297 36  80000001 21
```

ospf set area

Purpose

Sets the parameters for an OSPF area.

Format

ospf set area <*area-num*> **[stub] [stub-cost** <*num*>**] [authentication-method none | simple | md5]**

Mode

Configure

Parameters

<*area-num*>
The Area ID.

stub
Makes this Area a stub area.

stub-cost <*num*>
Specifies the cost to be used to inject a default route into the area. Specify a number from 0 – 65535.

authentication-method none | simple | md5
Specifies the authentication method used within the area. Specify one of the following:

none Does not use authentication.

simple Uses a simple string (password) up to 8 characters in length for authentication. If you chose this authentication method, then you should also specify a key-chain identifier using the key-chain option.

md5 Uses the MD5 algorithm to create a crypto-checksum of an OSPF packet and an authentication key of up to 16 characters.

Restrictions

None.

ospf set ase-defaults

Purpose

Sets the defaults used when importing OSPF ASE routes into the routing table and exporting routes from the routing table into OSPF ASEs.

Format

ospf set ase-defaults [preference <*num*>**] [cost** <*num*>**] [type** <*num*>**] [inherit-metric]**

Mode

Configure

Parameters

preference <*num*>
Specifies the preference of OSPF ASE routes. Specify a number between 0 and 255.

cost <*num*>
Specifies the cost used when exporting non-OSPF route into OSPF as an ASE. Specify a number from 0 – 65535.

type <*num*>
Specifies the ASE type. Routes exported from the routing table into OSPF default to becoming type 1 ASEs. You can change the default using the **type** option. You also can override the type in OSPF export policies. Specify either 1 or 2.

inherit-metric
Allows an OSPF ASE route to inherit the metric of the external route when no metric is specified on the export. A metric specified with the export command takes precedence. The cost specified in the default is used if you do not specify **inherit-metric**.

Restrictions

None.

ospf set export-interval

Purpose

Specifies the interval at which ASE LSAs will be generated and flooded into OSPF. The default is once per second.

Format

ospf set export-interval <*num*>

Mode

Configure

Parameters

<*num*> The interval in seconds. Specify a number equal to or greater than 1. The default is 1 (once per second).

Restrictions

None.

ospf set export-limit

Purpose

Specifies how many ASEs will be generated and flooded in each batch.

Format

ospf set export-limit <*num*>

Mode

Configure

Parameters

<*num*> The export limit. Specify a number equal to or greater than 1. The default is 100.

Restrictions

None.

ospf set interface

Purpose

Sets parameters for an OSPF interface.

Format

ospf set interface <*name-or-IPaddr*> **| all**
[state disable | enable] [cost <*num*>**] [no-multicast]**
[retransmit-interval <*num*>**] [transit-delay** <*num*>**]**
[priority <*num*>**] [hello-interval** <*num*>**]**
[router-dead-interval <*num*>**] [poll-interval** <*num*>**]**
[key-chain <*num-or-string*>**]**

Mode

Configure

Parameters

<*name-or-IPaddr*> **| all**
The OSPF interface for which you are setting OSPF parameters.

state disable | enable
Enables or disables OSPF on the interface.

cost <*num*>
The cost associated with this interface. The cost of all interfaces that a packet must cross to reach a destination are added to get the cost to that destination. The default cost of the OSPF interface is 1, but another non-zero value may be specified. Specify a number from 0 – 65535.

no-multicast
Instructs the SSR not to send multicast packets to neighbors on point-to-point interfaces.

retransmit-interval <*num*>
The number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface. Specify a number equal to or greater than 1. The default is 5.

transit-delay <*num*>
The estimated number of seconds required to transmit a link state update over this interface. Transit delay takes into account transmission and propagation delays and must be greater than 0. Specify a number equal to or greater than 1. The default is 1.

priority <*num*>
A number between 0 and 255 specifying the priority for becoming the designated router on this interface. When two routers attached to a network both attempt to become the designated router, the one with the higher priority wins. A router whose router priority is set to 0 is ineligible to become designated router. Specify a number from 0 – 255. The default is 1.

hello-interval <*num*>
The length of time, in seconds, between hello packets that the router sends on this interface. Specify a number from 0 – 255. The default is 10 for broadcast interfaces and 30 for point-to-point and other non-broadcast interfaces.

router-dead-interval <*num*>
The number of seconds not hearing a router's Hello packets before the router's neighbors will declare it down. Specify a number from 0 – 255. The default is 4 times the value of the hello interval.

poll-interval <*num*>
Before adjacency is established with a neighbor, OSPF packets are sent periodically at the specified poll interval. Specify a number equal to or greater than 1. The default value for this option is 120 seconds.

key-chain <*num-or-string*>
The identifier of the key-chain containing the authentication keys.

Restrictions

None.

ospf set monitor-auth-method

Purpose

You can query the OSPF state using the OSPF-Monitor utility. This utility sends non-standard OSPF packets that generate a text response from OSPF. By default these requests are not authenticated. If you specify an authentication key, the incoming requests must match the specified authentication key.

Format

ospf set monitor-auth-method none | simple | md5

Mode

Configure

Description

This section contains a fuller description of what the command does.

Parameters

authentication-method none | simple | md5
The authentication method used within the area. Specify one of the following:

none	Does not use authentication.
simple	Uses a simple string (password) up to 16 characters in length for authentication. If you chose this authentication method, then you should also specify a key-chain identifier using the key-chain option.
md5	Uses the MD5 algorithm to create a crypto-checksum of an OSPF packet and an authentication key of up to 16 characters.

Restrictions

None.

ospf set trace-options

Purpose

Sets various OSPF trace options.

Format

ospf set trace-options lsa-build | spf | lsa-transmit | lsa-receive

ospf set trace-options hello | dd | request | lsu | ack [detail] [send] [receive]

Mode

Configure

Parameters

lsa-build	Traces Link State Advertisement Creation.
spf	Traces Shortest Path First (SPF) calculations.
lsa-transmit	Traces Link State Advertisement (LSA) transmission.
lsa-receive	Traces Link State Advertisement (LSA) reception.
hello	Traces OSPF hello packets that are used to determine neighbor reachability.
dd	Traces OSPF Database Description packets that are used in synchronizing OSPF databases.
request	Traces OSPF Link State Request packets which are used in synchronizing OSPF databases.
lsu	Traces OSPF Link State Update packets which are used in synchronizing OSPF databases.
ack	Traces OSPF Link State Ack packets which are used in synchronizing OSPF databases.
detail	Shows detailed information about OSPF packets.

send	Shows OSPF packets sent by the router.
receive	Shows OSPF packets received by the router.

Restrictions

None.

ospf set virtual-link

Purpose

Sets the parameters for an OSPF virtual link.

Format

ospf set virtual-link <*number-or-string*>
[state disable | enable] [cost <*num*>**] [no-multicast] [retransmit-interval** <*num*>**]**
[transit-delay <*num*>**] [priority** <*num*>**] [hello-interval** <*num*>**]**
[router-dead-interval <*num*>**] [poll-interval** <*num*>**]**

Mode

Configure

Parameters

<*number-or-string*>
The identifier for this virtual link.

state disable | enable
Enables or disables the virtual link.

cost <*num*>
The cost associated with this virtual link. The cost of all interfaces that a packet must cross to reach a destination are added to get the cost to that destination. The default cost of the OSPF interface is 1, but another non-zero value may be specified. Specify a number from 0 – 65535.

no-multicast
Instructs the SSR to not send multicast packets to neighbors on point-to-point virtual links.

retransmit-interval <*num*>
The number of seconds between link state advertisement retransmissions for adjacencies belonging to this virtual link. Specify a number equal to or greater than 1.

transit-delay <*num*>
The estimated number of seconds required to transmit a link state update over this virtual link. Transit delay takes into account transmission and propagation delays and must be greater than 0. Specify a number equal to or greater than 1.

priority <*num*>
A number between 0 and 255 specifying the priority for becoming the designated router on this virtual link. When two routers attached to a network both attempt to become the designated router, the one with the higher priority wins. A router whose router priority is set to 0 is ineligible to become designated router. Specify a number from 0 – 255.

hello-interval <*num*>
The length of time, in seconds, between hello packets that the router sends on this virtual link. Specify a number from 0 – 255. The default is 60 seconds.

router-dead-interval <*num*>
The number of seconds not hearing a router's Hello packets before the router's neighbors will declare it down. Specify a number from 0 – 255. The default value for this parameter is 4 times the value of the **hello-interval** parameter

poll-interval <*num*>
Before adjacency is established with a neighbor, OSPF packets are sent periodically at the specified poll interval. Specify a number from 0 – 255. The default is 120 seconds.

Restrictions

None.

ospf show

Purpose

Show OSPF information.

Format

ospf show <*option-list*>

Mode

Enable

Parameters

<*option-list*>
Specifies the OSPF information you want to display. Specify one or more of the following:

all	Displays all OSPF tables.
globals	Displays OSPF globals.
timers	Displays OSPF timers.
areas	Displays OSPF areas.
interfaces	Displays OSPF interfaces.
next-hop-list	Displays valid next hop entries.
import-policies	Displays OSPF import policies.
export-policies	Displays OSPF export policies.
statistics	Displays OSPF statistics.
errors	Displays OSPF errors.
virtual-links	Displays OSPF virtual links.
summary-asb	Displays OSPF border routes.
AS-external-LDSB	Displays OSPF Autonomous System external link states.
exported-routes	Displays routes redistributed into OSPF.

Note: The **areas**, **virtual-links**, **summary-asb**, **AS-external-LDSB**, and **exported-routes** options can be used with the following display options:

to file Saves output in the file `/gatedtrc/gated.dmp`.

to terminal Displays output on the console. This is the default.

ospf start|stop

Purpose

Start or stop the OSPF protocol. OSPF is disabled by default on the SSR.

Format

ospf start | stop

Mode

Configure

Parameters

start Starts OSPF.

stop Stops OSPF.

Restrictions

None.

Chapter 36

ping Command

The **ping** command tests connection between the SSR and an IP host.

Format

ping <*hostname-or-IPaddr*> **packets** <*num*> **size** <*num*> **wait** <*num*> **[flood] [dontroute]**

Mode

User or Enable

Description

The **ping** command test connection between the SSR and an IP host. The ping command sends ICMP echo packets to the host you specify.

- If the packets reach the host, the host sends a ping response to the SSR and the CLI displays messages stating that the host can be reached.
- If the host does not respond, the SSR assumes the host cannot be reached from the SSR and the CLI display messages stating that the host did not reply.

Parameters

<*hostname-or-IPaddr*>
The host name or IP address you want to ping.

packets <*num*>
The number of ping packets you want to send. The default is 1.

size <*num*>
The packet size. For Ethernet, specify a number from 0 – 1364.

wait <*num*>
The number of seconds the SSR will wait for a positive response from the host before assuming that the host has not responded. The default is 1.

flood
Causes the SSR to send a new ping request as soon as a ping reply is received. If you do not specify the **flood** option, the SSR waits to send a new request. The amount of time the SSR waits is specified by the **wait** option.

dontroute
Restricts the ping to locally attached hosts.

Restrictions

If you enter this command from the User mode, the only parameter you can use is <*hostname-or-IPaddr*>. To use any of the other parameters, you must be in Enable mode.

Chapter 37

port Commands

The port commands set and display the following parameters:

- Port state (enabled or disabled)
- Bridging status (flow-based or address-based)
- Port operating mode (half duplex or full duplex)
- Port speed for the 10/100 ports (10-Mbps or 100-Mbps)
- Port mirroring (used for analyzing network traffic)
- Port shut down if broadcast threshold is reached

Command Summary

Table 26 lists the port commands. The sections following the table describe the command syntax.

Table 26. port commands

port bmon <*port-list*> **rate** <*number*> **duration** <*number*> **shutdown** <*number*>
port disable <*port-list*>
port flow-bridging <*port-list*> **| all-ports**
port mirroring to <*port*> **cpu-port-traffic	traffic-from** [<*port*> **	any] traffic-to** [<*slot*> **	any]**
port set [<*port-list*> **| all-ports] [duplex full	half]** **[speed 10Mbps	100Mbps |** <*number*>**] [auto-negotiation on	off]** **[hash-mode m0	m1	m2	m3] [wan encapsulation frame-relay	ppp] [clock** <*clock-source*>**]**

Table 26. port commands (Continued)

port show bmon
port show bridging-status <*port-list*> | **all-ports**
port show port-status <*port-list*> | **all-ports**
port show stp-info <*port-list*> | **all-ports**
port show vlan-info <*port-list*> | **all-ports**
port show mirroring-status <*slot*> | **all-slots**

port bmon

Purpose

Monitor broadcast traffic on a port.

Format

port bmon <*port-list*> **rate** <*number*> **duration** <*number*> **shutdown** <*number*>

Mode

Configure

Description

The **port bmon** command allows you to monitor the broadcast traffic on one or more ports and shut down a port if its broadcast traffic reaches and sustains a certain rate limit for a specified length of time. You can specify the duration of the port shut down.

Parameters

port <*port-list*>
Specifies the ports that you are monitoring for broadcasts.

rate <*number*>
The rate limit, in Kpkts per second, which will trigger a port shut down if the rate is sustained for the specified duration. Values can be from 1-1000. The default value is 10.

duration <*number*>
The number of seconds that the specified rate limit is sustained, after which the port will be shut down. Values can be from 1-3600. The default value is 1.

shutdown <*number*>
The number of seconds that the port will be shut down if the rate threshold is reached. Values can be from 60-36000. The default value is 300.

Restrictions

None.

Examples

To monitor broadcast traffic on port et.1.3 and shut it down for 5 minutes if the rate of 10,000 packets per second is sustained for 1 second:

```
ssr(config)# port bmon et.1.3
```

To monitor broadcast traffic on port et.1.3 and shut it down for 3 minutes if the rate of 25,000 packets per second is sustained for 5 seconds:

```
ssr(config)# port bmon et.1.3 rate 25 duration 5 shutdown 180
```

port disable

Purpose

Disable a port.

Format

port disable <*port-list*>

Mode

Configure

Description

The **port disable** command disables the specified ports. Disabled ports do not send or receive any traffic. You might want to disable unused ports to prevent network users from inadvertently or unscrupulously connecting to unoccupied but enabled ports on the SSR.

Parameters

port <*port-list*> Specifies the ports you are disabling.

Restrictions

None.

Examples

To disable port et.1.3 on the SSR:

```
ssr(config)# port disable et.1.3
```

To disable ports 1 through 5 on the Ethernet line card in slot 3 of the SSR chassis:

```
ssr(config)# port disable et.3.1-5
```

port flow-bridging

Purpose

Set ports to use flow-based bridging.

Format

port flow-bridging <*port-list*> | **all-ports**

Mode

Configure

Description

The **port flow-bridging** command changes the specified ports from using address-based bridging to using flow-based bridging. A port can use only one type of bridging at a time.

Each port has an L2 lookup table where MAC address or flows are stored.

- If the port is configured for address-based bridging (default), each L2 table entry consists of a MAC address and a VLAN ID.
- If the port is configured for flow-based bridging, each L2 table entry consists of a source MAC address, a destination MAC address, and a VLAN ID.

Suppose that a port on the SSR is connected to a hub that is connected to three workstations, A, B, and C. If each workstation is talking to one another and sending broadcast traffic, the L2 table on the SSR's port would contain the following entries for the workstations. Assume that the VLAN ID is "1" for all entries.

If the ports are configured for address-based bridging:

- MAC address A
- MAC address B
- MAC address C
- MAC broadcast address

If the ports are configured for flow-based bridging:

- MAC addresses A->B

- MAC addresses B->A
- MAC addresses B->C
- MAC addresses A->C
- MAC addresses C->A
- MAC addresses C->B
- MAC addresses A->broadcast
- MAC addresses B->broadcast
- MAC addresses C->broadcast

Parameters

<*port-list*> | **all-ports** Specifies the ports you are changing to flow-based bridging. The keyword **all-ports** changes all the ports on the SSR to flow-based bridging.

Restrictions

None.

Examples

To configure Ethernet port et.3.7 for flow-based bridging:

```
ssr(config)# port flow-bridging et.3.7
```

port mirroring

Purpose

Mirror traffic to a port for external analysis.

Format

port mirroring to <*port*> **cpu-port-traffic | traffic-from [**<*port*>**| any]**
traffic-to [<*slot*>**| any]**

Mode

Configure

Description

The **port mirroring** command mirrors the type of traffic you specify to a port. By attaching a protocol analyzer to the port, you can observe and analyze the mirrored traffic.

Parameters

<*port*>
Specifies the port to which you want to send the mirrored traffic. Attach your protocol analyzer to this port.

cpu-port-traffic
Mirrors traffic forwarded out by the Control Module. If you specify this option, you cannot specify the **traffic-from** or **traffic-to** options.

traffic-from [<*port*>**| any]**
Mirrors all traffic coming from the specified port. If you specify this option, you must also specify the **traffic-to** option.

traffic-to [<*port*>**| any]**
Mirrors traffic sent to the specified slot. The keyword **any** mirrors traffic sent to any of the SSR slots that contain line cards. If you specify this option, you must also specify the **traffic-to** option. To mirror traffic from the Control Module, use the **cpu-port-traffic** option.

Restrictions

Note the following restrictions:

- Unless you are mirroring the traffic form the Control Module, you must specify either an input port or an output slot.
- You cannot specify the **any** keyword with both the **traffic-from** and **traffic-to** options at the same time.
- None of the ports on the slot containing the protocol analyzer port can send or receive traffic while port mirroring is taking place. When a port is selected to receive mirrored traffic, none of the other ports on the line card can be used for normal traffic. For this reason, the protocol analyzer port cannot be on the same slot (line card) as the mirrored port(s).
- Do not configure an interface on the protocol analyzer port.
- Port Mirroring is not currently supported for WAN ports.

Examples

To copy traffic coming from port et.3.1 and going to any slot, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
ssr(config)# port mirroring to et.1.1 traffic-from et.3.1 traffic-to any
```

To copy traffic coming from any port and going to slot 4, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
ssr(config)# port mirroring to et.1.1 traffic-from any traffic-to 4
```

To capture all traffic going to and from the Control Module, enter the following command. The copied traffic is sent to port et.1.1, to which the protocol analyzer is attached.

```
ssr(config)# port mirroring to et.1.1 cpu-port-traffic
```

port set

Purpose

Set port operating mode and port speed.

Format

port set [<*port-list*> | **all-ports**] [**duplex full** | **half**]
[**speed 10Mbps** | **100Mbps** | <*number*>] [**auto-negotiation on** | **off**]
[**hash-mode m0** | **m1** | **m2** | **m3**] [**wan-encapsulation frame-relay** | **ppp**] [**ifg** <*number*>]
[**input-encapsulation forced-ethernet_ii**] [**link-timer** <*number*>] [**clock** <*clock-source*>]

Mode

Configure

Description

Depending on the media type of a port, the **port set** command lets you set various parameters of each port.

For 10/100-Mbps Ethernet, you can set the following:

- Operating mode (half-duplex or full-duplex).
- Port speed (10-Mbps or 100-Mbps). This parameter applies only to ports on the 10/100 line cards.
- Hash mode

Note: By default, all ports use autosensing to detect the operating mode and speed of the network segment to which they are connected. If you use this command to set a port parameter, the setting disables autosensing for that parameter on the port. For example, if you set the speed of a segment to 10-Mbps, that segment no longer uses autosensing for the port speed and will always attempt to operate at 10-Mbps.

For Gigabit Ethernet, you can set the following:

- Auto-negotiation
- Hash mode

For WAN ports, you can set the following:

- Wan-encapsulation (either frame-relay or ppp) and clock source (HSSI ports only)
- Speed (in Megabits per second)

Note: "Duplex", "autonegotiation", and "hash mode" are not applicable parameters for WAN interfaces.

Parameters

<*port-list*> | **all-ports**
Specifies the ports. The **all-ports** keyword applies the settings you select to all the SSR ports.

duplex full | half
Sets the operating mode to half duplex or full duplex. This option is valid for 10/100 Mbps Ethernet only.

speed 10Mbps | 100Mbps
Sets the port speed to 10-Mbps or 100-Mbps. This option is valid for 10/100 Mbps Ethernet only.

auto-negotiation on | off
Turns on or off auto-negotiation for Gigabit Ethernet.

hash-mode m0 | m1 | m2 | m3
Sets the Layer 2 hash mode for this port. Assuming a MAC address of the value 0011:2233:4455, the following describes the various hash modes:

- **m0** – 0011:2233:4455
- **m1** – 0011:2233:5544
- **m2** – 0011:3322:4455 (default hash mode)
- **m3** – 1100:2233:4455

wan-encapsulation frame-relay | ppp
Sets the encapsulation for the WAN port to either frame-relay or ppp.

ifg <*number*>
Changes the interframe gap (IFG) for the port by the amount specified by <*number*>. The <*number*> is a delta value in 40-nanosecond units for the IFG. Possible values for <*number*> are -12 through 64.

input-encapsulation forced-ethernet_ii
Changes the interpretation of the input MAC encapsulation to Ethernet II.

link-timer <*number*>
Sets the auto-negotiation link timer to the number of milliseconds specified by

<*number*>. The <*number*> is a value between 0 and 20. This option is valid for Gigabit ports only.

clock <*clock-source*>

Sets the clock source. This parameter is applicable only when the **wan-encapsulation** parameter is specified for a HSSI port that will be connected back-to-back with a HSSI port on another router. The <*clock-source*> is one of the following values:

external-clock	External transmit clock (DCE provided)
internal-clock-51mh	Internal transmit clock at 51.84 Mhz
internal-clock-25mh	Internal transmit clock at 25.92 Mhz
external-rx-clock	External receive clock for transmit clocking

Restrictions

For 10/100 Mbps Ethernet, you must set both the operating mode and the speed. You cannot set one without setting the other. For Gigabit Ethernet, you can only turn on or off auto-negotiation. You cannot set the speed or duplex for Gigabit modules.

Examples

To configure port et.1.5 to be 10 Mbps and half duplex:

```
ssr(config)# port set et.1.5 speed 10mbps duplex half
```

To turn off auto-negotiation for the Gigabit port gi.4.2:

```
ssr(config)# port set gi.4.2 auto-negotiation off
```

To set the Layer 2 hash mode for all ports to m0:

```
ssr(config)# port set all-ports hash-mode m0
```

To set the speed for a HSSI ppp WAN port located on port 1 of slot 3:

```
ssr(config)# port set hs.3.1 wan-encapsulation ppp speed 45000000
```

To set an internal clock source (25.92 Mhz) for a HSSI ppp WAN port located on port 1 of slot 3:

```
ssr(config)# port set hs.3.1 wan-encapsulation ppp speed 45000000 clock
internal-clock-25mh
```

To set the speed for a serial frame relay WAN port located at port 4 of slot 2, VC 100:

```
ssr(config)# port set se.2.4.100 wan-encapsulation frame-relay speed
    1500000
```

To increase the interframe gap for port et.1.1 by 400 nanoseconds (10 * 40ns):

```
ssr(config)# port set ifg et.1.1 ifg 10
```

port show bmon

Purpose

Display broadcast monitoring information for SSR ports.

Format

port show bmon [config][detail][port <*port list*>**][stats]**

Mode

Enable

Description

The **port show bmon** command lets you display broadcast monitoring information for SSR ports.

Parameters

If no parameters are specified, the current states of all ports are displayed.

config	Displays configuration information for broadcast monitoring.
detail	Displays all information for broadcast monitoring.
port <*port-list*>	Specifies the ports for which you want to display information.
stats	Displays statistics information for broadcast monitoring.

Restrictions

None.

Example

To display the state of ports with broadcast monitoring:

```
ssr# port show bmon
Port: et.1.1 State: On

Port: et.6.8 State: ShutDn Expire: 39 (sec)

Port: et.7.8 State: On
```

The above example shows three ports, with the port et.6.8 shut down for 39 seconds.

To display broadcast monitoring configuration values set for the ports:

```
ssr# port show bmon config
Port: et.1.1 Rate (Kpps): 10 Burst (sec): 1 Shutdown (sec):300

Port: et.6.8 Rate (Kpps): 10 Burst (sec): 5 Shutdown (sec):60

Port: et.7.8 Rate (Kpps): 2 Burst (sec): 2 Shutdown (sec):60
```

In the above example, port et.1.1 has been configured with default values.

To display broadcast monitoring statistics for the ports:

```
ssr# port show bmon stats
Port: et.1.1 Current Broadcast Rate (Kpps): 0.000

Port: et.6.8 Burst at port shutdown (Kpps): 10.032
ShutDn Count: 2

Port: et.7.8 Current Broadcast Rate (Kpps): 0.000
```

In the above example, the current broadcast traffic on et.1.1 and et.7.8 is zero. The port et.6.8 is currently shut down and it shows a burst of 10.032K packets per second at its shut down. This port has been shut down twice because of excessive broadcast traffic.

To show broadcast monitoring details for the ports:

```
ssr# port show bmon detail
Port: et.1.1 Rate (Kpps): 10 Burst (sec): 1 Shutdown (sec):300
State: On
Current Broadcast Rate (Kpps): 0.000

Port: et.6.8 Rate (Kpps): 10 Burst (sec): 5 Shutdown (sec):60
State: ShutDn Expire: 39 (sec)
Burst at port shutdown (Kpps): 10.032
ShutDn Count: 2

Port: et.7.8 Rate (Kpps): 2 Burst (sec): 2 Shutdown (sec):60
State: On
Current Broadcast Rate (Kpps): 0.000
```

The above example shows configuration, state, and statistics information.

port show bridging-status

Purpose

Display the bridging status of SSR ports.

Format

port show bridging-status <*port-list*> | **all-ports**

Mode

Enable

Description

The **port show bridging-status** command lets you display bridging-status information for SSR ports.

Parameters

<*port-list*>	**all-ports**	Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the SSR ports.

Restrictions

None.

Example

To display the bridging status for all available ports:

```
ssr# port show bridging-status all-ports
Port         Mgmt Status  phy-state     link-state Bridging Mode
----         -----------  ---------     ---------- -------------
et.4.1       No Action    Disabled      Link Down  Address
et.4.2       No Action    Disabled      Link Down  Address
et.4.3       No Action    Forwarding    Link Up    Address
et.4.4       No Action    Disabled      Link Down  Address
et.4.5       No Action    Disabled      Link Down  Address
et.4.6       No Action    Forwarding    Link Up    Address
et.4.7       No Action    Disabled      Link Down  Address
et.4.8       No Action    Disabled      Link Down  Address
```

port show port-status

Purpose

Display various information about specified ports.

Format

port show port-status <*port-list/SmartTRUNK-list*> **| all-ports | all-smarttrunks**

Mode

Enable

Description

The **port show port-status command** lets you display port-status information for SSR ports or SmartTRUNKs.

Parameters

<*port-list/SmartTRUNK-list*> **| all-ports | all-smarttrunks**
Specifies the LAN/WAN ports or SmartTRUNKs for which you want to display status information. The **all-ports** keyword displays information for all the SSR ports. The **all-smarttrunks** keyword displays information for all SmartTRUNKs.

Restrictions

This command does not show Virtual Circuit (VC) information. To see the state of sub-interfaces, you need to use the appropriate facility command, such as the **frame-relay show stats** command.

Example

To display the port status for all ports on Ethernet module 1 (et.1):

```
ssr# port show port-status et.1.*

Flags: M - Mirroring enabled  S - SmartTRUNK port

                                                                Link  Admin
Port     Port Type              Duplex Speed      Negotiation  State State Flags
----     ---------              ------ -----      -----------  ----- ----- -----
et.1.1   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.2   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.3   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.4   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.5   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.6   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.7   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
et.1.8   10/100-Mbit Ethernet   Half   10 Mbits   Manual       Up    Up
```

port show stp-info

Purpose

Display Spanning Tree (STP) information for SSR ports.

Format

port show stp-info <*port-list*> | **all-ports**

Mode

Enable

Description

The **port show stp-info** command lets you display Spanning-Tree information for SSR ports.

Parameters

<*port-list*>	**all-ports**	Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the SSR ports.

Restrictions

None.

Example

To display the spanning tree information for all available ports:

```
ssr# port show stp-info all-ports
Designated
Port      Priority  Cost   STP       State      Designated-Bridge     Port
----      --------  ----   ---       -----      -----------------  ----------
et.1.1       128    00100  Enabled   Listening  8000:00e063111111     80 01
et.1.2       128    00100  Enabled   Listening  8000:00e063111111     80 02
et.1.3       128    00100  Enabled   Listening  8000:00e063111111     80 03
et.1.4       128    00100  Enabled   Listening  8000:00e063111111     80 04
et.1.5       128    00100  Enabled   Listening  8000:00e063111111     80 05
et.1.6       128    00100  Enabled   Listening  8000:00e063111111     80 06
et.1.7       128    00100  Enabled   Listening  8000:00e063111111     80 07
et.1.8       128    00100  Enabled   Listening  8000:00e063111111     80 08
```

port show vlan-info

Purpose

Display VLAN information for SSR ports.

Format

port show vlan-info <*port-list*> | **all-ports**

Mode

Enable

Description

The **port show vlan-info** command lets you display VLAN information about SSR ports.

Parameters

<*port-list*>	**all-ports** Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the SSR ports.

Restrictions

None

Example

To display the VLAN information for all available ports:

```
ssr# port show vlan-info all-ports
Port         Access Type     IP VLANs     IPX VLANs    Bridging VLANs
------       ---------       ---------    --------     --------------
et.4.1       access          DEFAULT      DEFAULT      DEFAULT
et.4.2       access          DEFAULT      DEFAULT      DEFAULT
et.4.3       access          DEFAULT      DEFAULT      DEFAULT
et.4.4       access          DEFAULT      DEFAULT      DEFAULT
et.4.5       access          DEFAULT      DEFAULT      DEFAULT
et.4.6       access          DEFAULT      DEFAULT      DEFAULT
et.4.7       access          DEFAULT      DEFAULT      DEFAULT
et.4.8       access          DEFAULT      DEFAULT      DEFAULT
```

port show mirroring-status

Purpose

Show the port mirroring status for slots in the SSR chassis.

Format

port show mirroring-status <*slot*> | **all-slots**

Mode

Enable

Description

The **port show mirroring-status** command shows the following port mirroring status information for the specified chassis slots:

- Whether port mirroring is enabled
- The ports or slots that are being mirrored
- The mirroring mode (input port, output slot, or both)

Parameters

<*slot*> | **all-slots**	Specifies the chassis slots for which you want to display port mirroring status. The **all-slots** keyword displays port mirroring status for all the slots in the chassis.

Restrictions

None.

Examples

To display the port mirroring status for slot 5:

```
ssr(config)# port show mirroring-status 5
```

Chapter 38

port mirroring Command

Purpose

Apply port mirroring to one or more target ports on an SSR or to traffic specified by an ACL profile.

Format

port mirroring monitor-port <*port number*> **target-port** <*port list*> | **target-profile** <*acl name*>

Mode

Configure

Description

The **port mirroring** command allows you to monitor via a single port the activity of one or more ports on an SSR or the traffic that is specified by an ACL.

Parameters

monitor-port <*port number*>
The port you will use to monitor activity.

target-port *<port list>*

The port(s) for which you want to monitor activity. You can specify a single port or a comma-separated list of ports.

target-profile *<acl name>*

The name of the ACL that specifies the profile of the traffic that you want to monitor. The ACL must be a previously created IP ACL. The ACL may contain either **permit** or **deny** keywords. The **port mirroring** command only looks at the following ACL rule parameter values: protocol, source IP address, destination IP address, source port, destination port, and TOS.

Restrictions

Even though multiple target ports may be defined for a given SSR, only one monitor port may be defined. Also, Cabletron recommends that you monitor Gigabit ports through other Gigabit ports—you would almost certainly experience speed-inconsistency-related problems monitoring a Gigabit port through a 10Base-T or 100Base-TX port.

Known Problems

- Packets that are lost due to CRC and BUFFER_OVERFLOW errors are not mirrored to the monitor-port.
- In the example below, routed packets from source A to destination B on link 2 are seen as leaving src mac of SSR when port 1.2 is being monitored.

Examples

To mirror traffic on ethernet ports et.2.2-4 to port et1.2:

```
ssr(config)# port mirroring monitor-port et.1.2 target-port
    et.2.2 et.2.3 et.2.4
```

After configuring et.1.2 as a monitor-port, et.1.2 is unusable for any other function in the system. This is indicated by a A LINK_DOWN message. However, et.1.2 is capable of transmitting TX packets and its LED will be lit while in operation.

To mirror traffic that is specified by the profile in the ACL “101” to port et1.2:

```
ssr(config)# port mirroring monitor-port et.1.2 target-profile 101
```

Chapter 39

ppp Commands

The following commands allow you to define Point-to-Point Protocol (PPP) service profiles, and specify and monitor PPP High-Speed Serial Interface (HSSI) and standard serial ports.

Command Summary

Table 27 lists the PPP commands. The sections following the table describe the command syntax.

Table 27. ppp commands

ppp add-to-mlp <*mlp*> **port** <*port list*>
ppp apply service <*service name*> **ports** <*port list*>
ppp create-mlp <*mlp list*> **slot** <*number*>
ppp define service <*service name*> **[bridging enable | disable] [high-priority-queue-depth** <*number*>**] [ip enable | disable] [ipx enable | disable] [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]**
ppp restart lcp-ncp ports <*port list*>
ppp set mlp-encaps-format ports <*port list*> **[format short-format]**
ppp set mlp-frag-size ports <*port list*> **[size** <*number*>**]**
ppp set mlp-fragq-depth ports <*port list* > **qdepth** <*number-of-packets*>

Table 27. ppp commands (Continued)

ppp set mlp-orderq-depth ports <*port list* > **qdepth** <*number-of-packets*>
ppp set payload-compress [max-histories 0 | 1] [type stac] ports <*port list*>
ppp set payload-encrypt [type des-bis] transmit-key <*key*> **receive-key** <*key*> **ports** <*port list*>
ppp set peer-addr <*IP address*> <*IPX address*> **ports** <*port*>
ppp set ppp-encaps-bgd ports <*port list*>
ppp show mlp <*mlp list*> **| all-ports**
ppp show service <*service name*> **| all**
ppp show stats port <*port*> **[bridge-ncp] [ip-ncp] [link-status] [summary]**

ppp add-to-mlp

Purpose

Add PPP ports to an MLP bundle.

Format

ppp add-to-mlp <*mlp*> **port** <*port list*>

Mode

Configure

Description

The **ppp add-to-mlp** command allows you to add one or more PPP ports to a previously defined MLP bundle.

Parameters

<*mlp*>	The name of the previously defined MLP bundle.
<*port list*>	The WAN port(s) you want to add to the MLP bundle.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To add the port "hs.3.1" to the MLP bundle "mp.1":

```
ssr(config)# ppp add-to-mlp mp.1 port hs.3.1
```

ppp apply service

Purpose

Apply a pre-defined service profile to an interface.

Format

ppp apply service <*service name*> **ports** <*port list*>

Mode

Configure

Description

Issuing the **ppp apply service ports** command allows you to apply a previously defined service profile to a given PPP WAN port.

Parameters

<*service name*>	The name of the previously defined service you wish to apply to the given port(s) or interfaces.
<*port list*>	The port(s) to which you wish to apply the pre-defined service profile. You can specify a single port or a comma-separated list of ports.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To apply the service "s1" to slot 2, serial ports 1 and 2:

```
ssr(config)# ppp apply service s1 ports se.2.1 se.2.2
```

ppp create-mlp

Purpose

Create MLP bundles.

Format

ppp create-mlp <*mlp list*> **slot** <*number*>

Mode

Configure

Description

The **ppp create-mlp** command allows you to create one or more MLP bundles.

Parameters

<*mlp list*>	The name(s) of the MLP bundles you want to create. You can specify a single bundle or a comma-separated list of MLP bundles.
<*slot*>	The slot number for the MLP bundle(s).

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To create the MLP bundle “mp.1” for slot 1:

```
ssr(config)# ppp create-mlp mp.1 slot 1
```

ppp define service

Purpose

Define a service profile for WAN ports.

Format

ppp define service <*service name*> **[bridging enable | disable] [high-priority-queue-depth** <*number*>**] [ip enable | disable] [ipx enable | disable] [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]**

Mode

Configure

Description

The **ppp define service** command allows you to specify the following attributes for a newly created service profile:

- Activate and deactivate bridging, IP, and/or IPX for PPP WAN ports. If you do not specify any bridging, IP, or IPX protocols for PPP WAN ports, they are all activated by default. If you specify a bridging, IP, or IPX protocol, you *must* also explicitly define the behavior of the other two (i.e., **enabled** or **disabled**).
- The allowable PPP queue depth for high-, low-, and medium-priority items.
- Enable and disable the sending of LCP Echo Request messages. LCP Echo Requests and their corresponding LCP Echo Responses determine if a link to a peer is down.
- Enable and disable the use of LCP magic numbers. Magic numbers are used to help detect loopback conditions.
- The maximum allowable number of unanswered/improperly answered configuration requests before determining that the connection to the peer is lost.
- The maximum allowable number of negative-acknowledgment responses for a given interface before declaring an inability to converge.

- The maximum allowable unacknowledged terminate requests before determining that the peer is unable to respond.
- Activate or deactivate Random Early Discard (RED) for PPP ports.
- The maximum and minimum threshold values for RED high-, low-, and medium-priority traffic.

  In general, Cabletron recommends that the maximum threshold values be less than or equal to the respective high-, low-, or medium-priority queue depth. The minimum threshold values should be one-third of the respective maximum threshold.
- The number of seconds that will pass before a subsequent "resending" of the configuration request will be transmitted.
- Activate and deactivate RMON for PPP WAN ports. Before you can view RMON statistics such as Ethernet statistics and history for PPP WAN ports, RMON has to be activated.

Parameters

<service name>
The name you wish to assign to the newly created service profile.

bridging enable | disable
Specifying the **enable** keyword activates bridging for PPP WAN ports. Specifying the **disable** keyword deactivates bridging for PPP WAN ports.

high-priority-queue-depth *<number>*
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

ip enable | disable
Specifying the **enable** keyword activates IP for PPP WAN ports. Specifying the **disable** keyword deactivates IP for PPP WAN ports.

ipx enable | disable
Specifying the **enable** keyword activates IPX for PPP WAN ports. Specifying the **disable** keyword deactivates IPX for PPP WAN ports.

lcp-echo on | off
Specifying the **on** keyword enables the sending of LCP Echo Request messages. Specifying the **off** keyword disables the sending of LCP Echo Request messages. The sending of LCP Echo Requests is enabled by default.

lcp-magic on | off
Specifying the **on** keyword enables the use of LCP magic numbers. Specifying the **off** keyword disables the use of LCP magic numbers. The use of LCP magic numbers is enabled by default.

low-priority-queue-depth <*number*>
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

max-configure <*number*>
The maximum allowable number of unanswered requests. You can specify any number greater than or equal to 1. The default value is 10.

max-failure <*number*>
The maximum allowable number of negative-acknowledgment transmissions. You can specify any number greater than or equal to 1. The default value is 5.

max-terminate <*number*>
The maximum allowable number of unanswered/improperly answered connection-termination requests before declaring the link to a peer lost. You can specify any number greater than or equal to 1. The default value is 2.

med-priority-queue-depth <*number*>
The number of items allowed in the PPP queue. You can specify a number between 1 and 65,535. Cabletron recommends a value within the 5 - 100 item range. The default value is 20.

red on | off
Specifying the **on** keyword enables RED for PPP WAN ports. Specifying the **off** keyword disables RED for PPP WAN ports.

red-maxTh-high-prio-traffic <*number*>
The maximum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-maxTh-low-prio-traffic <*number*>
The maximum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-maxTh-med-prio-traffic <*number*>
The maximum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 12.

red-minTh-high-prio-traffic <*number*>
The minimum allowable threshold for high-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

red-minTh-low-prio-traffic <*number*>
The minimum allowable threshold for low-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

red-minTh-med-prio-traffic <*number*>
The minimum allowable threshold for medium-priority RED traffic. You can specify a number between 1 and 65,535. The default value is 4.

retry-interval <*number*>
The number of seconds between subsequent configuration request transmissions (the interval). You can specify any number greater than or equal to 1. The default value is 30.

rmon on | off
Specifying the **on** keyword enables RMON for PPP WAN ports. Specifying the **off** keyword disables RMON for PPP WAN ports.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To create a service profile named "pppserv4" with the following attributes:

- Bridging enabled
- IP and IPX enabled
- LCP Echo Requests disabled
- LCP magic numbers disabled
- RED disabled
- A retry interval of 20 seconds
- rmon enabled

then you would enter the following command line in Configure mode:

```
ssr(config)# ppp define service pppserv4 bridging enable ip enable ipx
    enable lcp-echo off lcp-magic off red off retry-interval 20 rmon on
```

ppp restart lcp-ncp

Purpose

Restart PPP LCP/NCP negotiation.

Format

ppp restart lcp-ncp ports <*port list*>

Mode

Enable

Description

The **ppp restart lcp-ncp** command allows you to reset and restart the LCP/NCP negotiation process for PPP WAN ports.

Parameters

<*port list*> The ports for which you would like to re-establish LCP/NCP negotiation.

Restrictions

This command line is available only for PPP WAN ports.

Example

To restart LCP/NCP negotiation on serial ports 1 and 2 of slot 4:

```
ssr# ppp restart lcp-ncp ports se.4.1 se.4.2
```

ppp set mlp-encaps-format

Purpose

Set MLP encapsulation format.

Format

ppp set mlp-encaps-format ports <*port list*> **[format short-format]**

Mode

Configure

Description

The **ppp set mlp-encaps-format** command allows you to specify the encapsulation format for MLP bundles. If this command is not configured, long format encapsulation is used for MLP bundles.

Parameters

<*port list*>
The MLP port(s) to which you want to apply the encapsulation format

format short-format
Specifies the use of short format for MLP encapsulation.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To specify short format encapsulation for the MLP bundles “mp.1” and “mp.4-7”:

```
ssr(config)# ppp set mlp-encaps-format ports mp.1 mp.4-7 format short-
format
```

ppp set mlp-frag-size

Purpose

Set the frame size under which no MLP fragmentation is needed.

Format

ppp set mlp-frag-size ports <*port list* > **[size** <*number*>**]**

Mode

Configure

Description

The **ppp set mlp-frag-size** command allows you to set the frame size under which no fragmentation is needed for transmission on the MLP bundle. The default size is 1500 bytes. Any frames that are less than the value set by the **ppp set mlp-frag-size** command are not fragmented. Any frames that are over the value are fragmented for transmission on the MLP bundle.

Parameters

<*port list*>	The MLP port(s) to which the frame size applies.
<*number*>	The size of the frame, in bytes, that are fragmented by MLP. The value can be between 64 and 1500, inclusive. The default value is 1500.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To specify that frames of 200 bytes or more are fragmented on the MLP bundles “mp.1” and “mp.4-7”:

```
ssr(config)# ppp set mlp-frag-size ports mp.1 mp.4-7 size 200
```

ppp set mlp-fragq-depth

Purpose

Set the depth of the MLP fragment queue.

Format

ppp set mlp-fragq-depth ports <*port list* > **qdepth** <*number-of-packets*>

Mode

Configure

Description

The **ppp set mlp-fragq-depth** command allows you to set the depth of the queue used by MLP to hold packet fragments for reassembly.

Parameters

<*port list*> The MLP port(s) to which the queue depth applies.

<*number-of-packets*>
The depth of the queue, in packets, to hold unassembled packet fragments. The value can be between 100 and 4000, inclusive. The default value is 1000.

Restrictions

Usage is restricted to MLP WAN ports only.

Example

To specify a queue depth of 2500 packets to hold fragments for reassembly on the MLP bundles “mp.1”::

```
ssr(config)# ppp set mlp-fragq-depth ports mp.1 size 2500
```

ppp set mlp-orderq-depth

Purpose

Set the depth of the MLP packet order queue.

Format

ppp set mlp-orderq-depth ports <*port list* > **qdepth** <*number-of-packets*>

Mode

Configure

Description

The **ppp set mlp-orderq-depth** command allows you to set the depth of the queue used by MLP to hold MLP packets for preserving the packet order.

Parameters

<*port list*> The MLP port(s) to which the queue depth applies.

<*number-of-packets*>
The depth of the queue, in packets, to hold MLP packets. The value can be between 100 and 4000, inclusive. The default value is 1000.

Restrictions

Usage is restricted to MLP WAN ports only.

Example

To specify a queue depth of 2500 packets to hold packets for reordering on the MLP bundles “mp.1”:

```
ssr(config)# ppp set mlp-orderq-depth ports mp.1 size 2500
```

ppp set payload-compress

Purpose

Enables packet compression for PPP ports.

Format

ppp set payload-compress [max-histories <*number*>**] [type stac] ports** <*port list*>

Mode

Configure

Description

The **ppp set payload-compress** command allows you to enable the Stacker payload compression. You can enable compression on a single port, an entire multilink PPP (MLP) bundle, or on individual ports that are members of a multilink PPP bundle. If this command is not configured, payload compression is not enabled.

Parameters

<*number*>
Specifies the maximum number of compression history buffers to be kept. You can specify either 0 or 1. Specifying 0 disables the keeping of any histories and each packet is individually compressed. Specifying 1 allows a history buffer to be kept, which may result in better compression. The default value is 1.

type stac
Specifies the Stacker (STAC LZS) compression algorithm. This is the default.

<*port list*>
The port(s) on which you want to enable payload compression. You can specify a single port or a comma-separated list of ports.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To enable LZS Stac payload compression on slot 4, on serial port 2:

```
ssr(config)# ppp set payload-compress port se.4.2
```

ppp set payload-encrypt

Purpose

Enables packet encryption for PPP ports.

Format

ppp set payload-encrypt [type des-bis] transmit-key <*key*> **receive-key** <*key*> **ports** <*port list*>

Mode

Configure

Description

The **ppp set payload-encrypt** command allows you to enable the encryption of packets using the DES-bis algorithm. You can enable encryption on a single port, an entire multilink PPP (MLP) bundle, or on individual ports that are members of an MLP bundle. If this command is not configured, payload encryption is not enabled.

Parameters

type des-bis
Specifies the DES-bis encryption algorithm. This is the default.

<*key*>
Specifies a 16-digit hexadecimal number for the encoding and decoding of the packets. The keys are themselves encrypted and stored in the active and startup configurations.

<*port list*>
The port(s) on which you want to enable payload encryption. You can specify a single port or a comma-separated list of ports.

Restrictions

Usage is restricted to PPP WAN ports only.

Example

To enable DES-bis payload encryption on slot 4, on serial port 2:

```
ssr(config)# ppp set payload-encrypt transmit-key 0x123456789abcdef0
receive-key 0xfedcba9876543210 port se.4.2
```

ppp set peer-addr

Purpose

Set the peer address in case that IPCP/IPXCP can't resolve the address.

Format

ppp set peer-addr <*IP address*> **ports** <*port*>

Mode

Configure

Description

Issuing the **ppp set peer-addr** command allows you to set the peer address if it can't be resolved by IPCP or IPXCP.

Parameters

<*address*>	The IP or IPX address you wish to use.
<*port*>	The port to which you wish to assign the address.

Restrictions

Usage is restricted to PPP port only.

Example

To assign an ip address 10.1.1.1/16 to slot 2, serial port 1:

```
ssr(config)# ppp set peer-addr ip-addr 10.1.1.1/16 ports se.2.1
```

ppp set ppp-encaps-bgd

Purpose

Force the ingress packets to be encapsulated in bridged format.

Format

ppp set ppp-encaps-bgd ports <*port list*>

Mode

Configure

Description

Issuing the **ppp set ppp-encaps-bgd** command allows you to use bridged format encapsulation on a given ppp port.

Parameters

<*port list*> The port(s) to which you wish to use bridged encapsulation. You can specify a single port or a comma-separated list of ports.

Restrictions

Usage is restricted to ppp port only.

Example

To force the bridged encapsulation to slot 2, serial ports 1 and 2:

```
ssr(config)# ppp ppp-encaps-bgd ports se.2.1 se.2.2
```

ppp show mlp

Purpose

Displays the PPP ports that have been added into an MLP bundle.

Format

ppp show mlp <*mlp list*> | **all-ports**

Mode

Enable

Description

The **ppp show mlp** command allows you to display information about one or more MLP bundles.

Parameters

<*mlp list*>	The name(s) of the MLP bundles on which you want information. You can specify a single bundle or a comma-separated list of MLP bundles.
all-ports	Displays information on all MLP ports.

Restrictions

None.

Example

To display the PPP ports for mp.1:

```
ssr# ppp show mlp mp.1
mp.1:
  Slot: 4
  PPP ports: se.4.1 se.4.3
```

ppp show service

Purpose

Displays PPP service profiles.

Format

ppp show service <*service name*> | **all**

Mode

Enable

Description

The **ppp show service** command allows you to display one or all of the available PPP service profiles.

Parameters

<*service name*>	The service profile you wish to display.
all	Displays all of the available PPP service profiles.

Restrictions

None.

Example

To display the available PPP service profiles named profile_4:

```
ssr# ppp show service profile_4
```

ppp show stats

Purpose

Displays bridge NCP, IP NCP, and link-status parameters.

Format

ppp show stats port <*port*> **[bridge-ncp] [ip-ncp] [link-status] [summary]**

Mode

Enable

Description

The **ppp show stats** command allows you to display parameters for bridge NCP, IP NCP, and link-status on PPP WAN ports. You can specify one, two, or three of the available parameter types.

Parameters

<*port*>	The PPP WAN port for which you wish to view bridge NCP, IP NCP, and/or link-status parameters.
bridge-ncp	Specifies that you wish to view bridging NCP parameters for the given port.
ip-ncp	Specifies that you wish to view IP NCP parameters for the given port.
link-status	Specifies that you wish to view link-status parameters for the given port.
summary	Specifies that you wish to view summarized display.

Restrictions

None.

Example

To display the available link-status and IP NCP parameters for the PPP WAN interface located at slot 4, port 1:

```
ssr# ppp show stats port se.4.1 ip-ncp link-status
```

Chapter 40

pvst Commands

The **pvst** commands let you display and change settings for a VLAN spanning tree.

Command Summary

Table 28 lists the **pvst** commands. The sections following the table describe the command syntax.

Table 28. stp commands

pvst create spanningtree vlan-name <*string*>
pvst enable port <*port-list*> **spanning-tree** <*string*>
pvst set bridging [**forward-delay** <*num*>] [**hello-time** <*num*>] [**max-age** <*num*>] [**priority** <*num*>] **spanning-tree** <*string*>
pvst set port <*port-list*> **priority** <*num*> **port-cost** <*num*> **spanning-tree** <*string*>
pvst show bridging-info spanning-tree <*string*>

pvst create spanningtree

Purpose

Create an instance of spanning tree for a particular VLAN.

Format

pvst create spanningtree vlan-name <*string*>

Mode

Configure

Description

The **pvst create spanningtree** command creates a spanning tree instance for a particular VLAN.

Parameters

vlan-name <*string*>
The name of the VLAN for which a new instance of spanning tree is to be created.

Restrictions

None.

pvst enable port spanning-tree

Purpose

Enable PVST on one or more ports on a particular spanning tree.

Format

pvst enable port <*port-list*> **spanning-tree** <*string*>

Mode

Configure

Description

The **pvst enable port** command enables STP on the specified port for the specified spanning tree.

Parameters

<*port-list*> The ports on which you are enabling STP. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

<*string*> The name of the spanning-tree instance. This name is the same as the VLAN name.

Note: For default VLAN, use **stp** commands.

Restrictions

For PVST, the spanning tree instance must have previously been created.

pvst set bridging spanning-tree

Purpose

Set STP bridging parameters for a particular VLAN.

Format

pvst set bridging [**forward-delay** <*num*>] [**hello-time** <*num*>] [**max-age** <*num*>] [**priority** <*num*>] **spanning-tree** <*string*>

Mode

Configure

Description

The **pvst set bridging spanning-tree** command lets you configure the following STP parameters for a particular VLAN:

- Bridging priority
- Hello time
- Maximum age
- Forward delay

Parameters

forward-delay <*num*>
Sets the STP forward delay for the SSR. The forward delay is measured in seconds. Specify a number from 4– 30. The default is 15.

hello-time <*num*>
Sets the STP hello time for the SSR. The hello time is measured in seconds. Specify a number from 1– 10. The default is 2.

max-age <*num*>
Sets the STP maximum age for the SSR. Specify a number from 6–40. The default is 20.

priority <*num*>
Sets the STP bridging priority for the SSR. Specify a number from 0 – 65535. The default is 32768

spanning-tree <*string*>
The name of the spanning-tree instance. This name is the same as the VLAN name.

Note: For default VLAN, use **stp** commands.

Restrictions

For PVST, the spanning tree instance must have previously been created.

Examples

To set the bridging priority of Spanning Tree for VLAN ip1 to 1:

```
ssr(config)# pvst set bridging priority 1 spanning-tree ip1
```

pvst set port spanning-tree

Purpose

Set PVST port priority and port cost for ports for a particular VLAN.

Format

pvst set port <*port-list*> **priority** <*num*> **port-cost** <*num*> **spanning-tree** <*string*>

Mode

Configure

Description

The **pvst set port** command sets the STP priority and port cost for individual ports for a particular VLAN.

Parameters

port <*port-list*>
The port(s) for which you are setting STP parameters. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

priority <*num*>
The priority you are assigning to the port(s). Specify a number from 0– 255. The default is 128.

port-cost <*num*>
The STP cost you are assigning to the port(s). Specify a number from 1– 65535. The default depends on the port speed: 1 for Gigabit (100-Mbps) ports, 10 for 100-Mbps ports, and 100 for 10-Mbps ports.

spanning-tree <*string*>
The name of the spanning-tree instance. This name is the same as the VLAN name.

Note: For default VLAN, use **stp** commands.

Restrictions

For PVST, the spanning tree instance must have previously been created.

pvst show bridging-info spanning-tree

Purpose

Display STP bridging information for a particular VLAN.

Format

pvst show bridging-info spanning-tree <*string*>

Mode

Enable

Description

The **pvst show bridging-info** command displays STP bridging information for a particular VLAN.

Parameters

spanning-tree <*string*>
The name of the spanning-tree instance. This name is the same as the VLAN name.

Note: For default VLAN, use **stp** commands.

Restrictions

For PVST, the spanning tree instance must have previously been created.

Chapter 41

qos Commands

The qos commands define and display Quality of Service (QoS) parameters. Use the command 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. Use the **qos set l2**, **qos set ip**, and **qos set ipx** commands to assign priorities for Layer-2, IP, and IPX traffic respectively.

Flows

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

A precedence from 1 – 7 is associated with each field in a flow. The SSR 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 are the default precedences of the fields:

- **IP** – destination port (1), destination address (2), source port (3), source IP address (4), TOS (5), interface (6), protocol (7).
- **IPX** – destination network (1), source network (2), destination node (3), source node (4), destination port (5), source port (6), interface (7).

Use the **qos precedence ip** and **qos precedence ipx** commands to change the default precedences.

Queuing Policies

You can use one of two queuing policies on the SSR:

- **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 throughout of control-priority traffic, and so on.
- **weighted fair queuing** – distributes priority throughput among the four priorities (control, high, medium, and low) based on percentages.

The SSR can use only one queuing policy at a time. The policy is used on the entire SSR. The default queuing policy is strict priority.

Command Summary

Table 29 lists the **qos** commands. The sections following the table describe the command syntax.

Table 29. qos commands

qos precedence [sip <*num*>**] [dip** <*num*>**] [srcport** <*num*>**] [destport** <*num*>**]** **[tos** <*num*>**] [protocol** <*num*>**] [intf** <*num*>**]**
qos precedence ipx [srcnet <*num*>**] [srcnode** <*num*>**]** **[srcport** <*num*>**] [dstnet** <*num*>**] [dstnode** <*num*>**] [dstport** <*num*>**] [intf** <*num*>**]**
qos set ip <*name*> <*priority*> <*srcaddr/mask*> **| any** <*dstaddr/mask*> **| any** <*srcport*> **| any** <*dstport*> **| any** <*tos*> **|** <*interface-list*> **| any** <*protocol*>
qos set ipx <*name*> <*priority*> <*srcnet*> **| any** <*srcmask*> **| any** <*srcport*> **| any** <*dstnet*> **| any** <*dstmask*> **| any** <*dstport*> **|** <*interface-list*> **| any**

Table 29. qos commands (Continued)

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*>
qos set queuing-policy weighted-fair
qos set weighted-fair control <*percentage*> **high** <*percentage*> **medium** <*percentage*> **low** <*percentage*>
qos show ip
qos show ipx
qos show l2 all-destination all-flow ports <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

qos precedence ip

Purpose

Set the precedence of the IP flow fields.

Format

qos precedence ip [sip <*num*>**] [dip** <*num*>**] [srcport** <*num*>**] [destport** <*num*>**] [tos** <*num*>**] [protocol** <*num*>**] [intf** <*num*>**]**

Mode

Configure

Description

The **qos precedence ip** command lets you set the QoS precedence for various flow fields in IP traffic. You can set a precedence from 1 – 7 for the following IP fields:

- IP source address
- IP destination address
- Source TCP or UDP port
- Destination TCP or UDP port
- Type of Service (TOS) for the packet
- Protocol (TCP or UDP)
- Incoming interface

The precedence 1 is the highest priority. IP interfaces or flow fields within IP packets that have a precedence of 1 are given first priority. The default priorities are as follows:

- destination port (1)
- destination address (2)
- source port (3)
- source IP address (4)
- TOS (5)

- interface (6)
- protocol (7).

Parameters

sip <*num*>
Specifies the precedence of the source address field in IP flows. Specify a precedence from 1 – 7.

dip <*num*>
Specifies the precedence of the destination address field in IP flows. Specify a precedence from 1 – 7.

srcport <*num*>
Specifies the precedence of the source port field in IP flows. Specify a precedence from 1 – 7.

dstport <*num*>
Specifies the precedence of the destination port field in IP flows. Specify a precedence from 1 – 7.

tos <*num*>
Specifies the precedence of the TOS field in IP flows. Specify a precedence from 1 – 7.

protocol <*num*>
Specifies the precedence of the transport layer protocol name field in IP flows. Specify a precedence from 1 – 7.

intf <*num*>
Specifies the precedence of the IP interface based on the interface's name. Specify a precedence from 1 – 7.

Restrictions

None.

Examples

To change the precedence for fields within IP flows from the default precedences listed above:

```
ssr(config)# qos precedence ip sip 3 dip 1 srcport 2 destport 4 tos 5
    protocol 6 intf 7
```

qos precedence ipx

Purpose

Set the precedence of the IPX flow fields.

Format

qos precedence ipx [srcnet <*num*>] **[srcnode** <*num*>**]** **[srcport** <*num*>**]** **[dstnet** <*num*>**]** **[dstnode** <*num*>**]** **[dstport** <*num*>**]** **[intf** <*num*>**]**

Mode

Configure

Description

The **qos precedence ipx** command lets you set the precedence of the following fields in IPX flows.

- Source network
- Source port
- Source node
- Destination network
- Destination node
- Destination port
- Incoming interface

You can set the precedence of the following fields from 1 – 7. The precedence 1 has the highest priority and 7 has the lowest. The default priorities are as follows:

- destination network (1)
- source network (2)
- destination node (3)
- source node (4)
- destination port (5)

- source port (6)
- interface (7).

Parameters

srcnet <*num*>
Specifies the precedence of the source network field in IPX flows. Specify a precedence from 1 – 7.

srcport <*num*>
Specifies the precedence of the source port field in IPX flows. Specify a precedence from 1 – 7.

srcnode <*num*>
Specifies the precedence of the source node field in IPX flows. Specify a precedence from 1 – 7.

dstnet <*num*>
Specifies the precedence of the destination network field in IPX flows. Specify a precedence from 1 – 7.

dstnode <*num*>
Specifies the precedence of the destination node field in IPX flows. Specify a precedence from 1 – 7.

dstport <*num*>
Specifies the precedence of the destination port field in IPX flows. Specify a precedence from 1 – 7.

intf <*num*>
Specifies the precedence of the IPX interface based on the interface's name. Specify a precedence from 1 – 7.

Restrictions

None.

Examples

To change the precedence for fields within IPX flows from the default precedences listed above:

```
ssr(config)# qos precedence ipx srcnet 1 srcnode 2 srcport
    dstnet 3 srcport 4 dstnode 5 dstport 6 intf 7
```

qos set ip

Purpose

Set a priority for an IP flow.

Format

qos set ip <*name*> <*priority*> [<*srcaddr/mask*> | **any**]
[<*dstaddr/mask*> | **any**] [<*srcport*> | **any**] [<*dstport*> | **any**] [<*tos*> | **any**] [<*interface-list*> | **any**]
[<*protocol*> | **any**] [<*tos-mask*> | **any**] [<*tos-precedence-rewrite*> | **any**] [<*tos-rewrite*> | **any**]

Mode

Configure

Description

The **qos set ip** command sets the priority for an IP flow based on the following fields in the flow:

- Flow name
- Source IP address and network mask
- Destination IP address and network mask
- Source port
- Destination port
- TOS
- Transport layer protocol (TCP or UDP)

You can set the priority of each field to control, low, medium, or high. The default is low.

Parameters

<*name*>
Specifies the IP flow name.

<priority>
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

control Assigns control priority to the IP flow parameters you have specified. This is the highest priority.

high Assigns high priority to the IP flow parameters you have specified.

medium Assigns medium priority to the IP flow parameters you have specified.

low Assigns low priority to the IP flow parameters you have specified. This is the default.

<srcaddr/mask> | **any**
Specifies the source IP address and network mask for which you are assigning a priority. You can specify the mask using the traditional IP address format ("255.255.0.0") or the CIDR format ("/16").

If you specify **any** instead of a network mask, the SSR assumes a wildcard "don't care" condition. If you do not specify a mask, then the SSR assumes a mask of 255.255.255.255. You cannot substitute the mask with the **any** keyword. The keyword **any** is for the entire *<srcaddr/mask>* pair.

<dstaddr/mask> | **any**
Specifies the destination IP address and network mask for which you are assigning a priority. The same requirements and restrictions for *<srcaddr/mask>* **apply to** *<dstaddr/mask>*.

If you specify **any** instead of a network mask, the SSR assumes a wildcard "don't care" condition. If you do not specify a mask, then the SSR assumes a mask of 255.255.255.255. You cannot substitute the mask with the **any** keyword. The keyword **any** is for the entire *<dstaddr/mask>* pair.

<srcport> | **any**
Specifies the source TCP or UDP port for which you are assigning a priority. Specify a port number from 1 – 65535 or **any** to allow any value.

<dstport> | **any**
Specifies the destination TCP or UDP port for which you are assigning a priority. Specify a port number from 1 – 65535 or **any** to allow any value.

<tos> | **any**
Specifies the TOS for which you are assigning a priority. Specify a number from 0– 15 or **any** to allow any value.

<interface-list> | **any**
Specifies one or more IP interface names for which you are assigning priority. If you specify a list, delimit the interface names with commas. Specify **any** to allow any IP interface name.

<*protocol*> | **any**
Specifies the transport layer protocol for which you are assigning priority. You can specify one of the following values:

tcp Assigns the priority parameters to the TCP protocol.

udp Assigns the priority parameters to the UDP protocol.

any Assigns the priority parameters to both the TCP and UDP protocols.

<*tos-mask*>
Specifies the mask that is used for the TOS byte. Specify a number from 1-255 or **any** to specify any TOS value. The default is 30.

<*tos-precedence-rewrite*>
Rewrites the precedence portion of the TOS field with a new value. Specify a number from 0-7 or **any** to specify any TOS value.

<*tos-rewrite*>
Rewrites the entire TOS field with a new value. Specify a number from 0-31 or **any** to specify any TOS value.

Note: If you set **any** for the TOS precedence rewrite and specify a value for <*tos-rewrite*>, then the precedence portion of the TOS field remains the same as in the packet, but the rest of the TOS field is rewritten. If you specify values for both <*tos-precedence-rewrite*> and <*tos-rewrite*>, then the precedence portion of the TOS field is rewritten to the new <*tos-precedence-rewrite*> number and the rest of the TOS field is rewritten to the new <*tos-rewrite*> number.

Restrictions

None.

Examples

The following command creates a flow called “flow1”. This flow provides a template for an IP packet with the IP address 1.1.1.1, network mask 255.255.0.0, destination address 2.2.2.2 (and implied destination mask 255.255.255.255). The flow includes source TCP/UDP port 3010, destination port 3000, a TOS of 15, the interfaces mls1 and mls2, and the TCP protocol as transport layer. This very explicit flow has the highest priority—control.

```
ssr(config)# qos set ip flow1 control 1.1.1.1/255.255.0.0 2.2.2.2 3010
    3000 15 mls1 mls2 tcp
```

qos set ipx

Purpose

Set a priority for an IPX flow.

Format

qos set ipx <*name*> <*priority*> [<*srcnet*> | **any**] [<*srcmask*> | **any**] [<*srcport*> | **any**] [<*dstnet*> | **any**] [<*dstmask*> | **any**] [<*dstport*> | **any**] [<*interface-list*> | **any**]

Mode

Configure

Description

The **qos set ipx** command lets you set the priority for an IPX flow based on the following fields in the flow:

- Flow name
- Source network
- Source network mask
- Source port
- Destination network
- Destination network mask
- Destination port

You can set the priority of each field to control, low, medium, or high. The default is low.

Parameters

<*name*>
Specifies the IPX flow name.

<*priority*>
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

control Assigns control priority to the IP flow parameters you have specified. This is the highest priority.

high Assigns high priority to the IP flow parameters you have specified.

medium Assigns medium priority to the IP flow parameters you have specified.

low Assigns low priority to the IP flow parameters you have specified. This is the default.

<*srcnet*> | **any**
Specifies the IPX source network and node address. Specify them in the following format: <*netaddr*>**.**<*macaddr*>; for example: `a1b2c3d4.aa:bb:cc:dd:ee:ff`.

If you specify **any** instead of a **.**<*macaddr*>, the SSR assumes a wildcard value. All MAC addresses are then valid.

<*srcmask*> | **any**
Specifies the IPX source network mask. Specify the mask in hexadecimal digits. If you do not specify a mask value and instead use the value **any**, the SSR internally sets the mask to FFFFFFFF.

<*srcport*> | **any**
Specifies a port number from 1 – 65535 or any to allow any value.

<*dstnet*> | **any**
Specifies the IPX destination network and node address. The same requirements and restrictions for <*dstaddr*> *apply to* <*srcaddr*>.

<*dstmask*> | **any**
Specifies the IPX destination network mask. Specify the mask in hexadecimal digits or **any** to allow any value.

<*dstport*> | **any**
Specifies a port number from 1 – 65535 or **any** to allow any value.

<*interface-list*> | **any**
If you specify a list, delimit the interface names with commas. Specify **any** to allow any IPX interface name.

Restrictions

None.

Examples

The following command creates an IPX flow called “abc”. This flow gives a high priority to IPX traffic on interface mls1 from network 12345678.00:01:00:00:00:00, mask 0000ff00, port 55 to network 22222222.02:00:00:00:00:00, mask 0000ff00, port 65.

```
ssr(config)# qos set ipx abc high 12345678.00:01:00:00:00:00 0000ff00 55
    22222222.02:00:00:00:00:00 0000ff00 65 mls1
```

qos set l2

Purpose

Configure priority for a Layer 2 flow.

Format

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*>

Mode

Configure

Description

The **qos set l2** command lets you set QoS priority on a Layer 2 flow. You can set priorities on the following fields in the flow:

- L2 flow name
- Source MAC address
- Destination MAC address
- VLAN ID
- Incoming port(s)

You can set the priority of each field in one of the following ways:

- The flow is assigned a priority within the switch. In this case you specify a priority of control, low, medium, or high. The default is low.
- The flow is assigned a priority within the switch, but in addition, if the exit ports are VLAN trunk ports, the flow is assigned an 802.1Q priority. In this case you specify a number from 1 – 7. The SSR maps the number to the four internal priorities as follows: 0 = low; 1, 2, or 3 = medium; 4, 5,or 6 = high; 7 = control.

Parameters

name <*name*>
Specifies the L2 flow name.

source-mac <*MACaddr*>
Specifies the L2 source MAC address. *Specify the MAC address in either of the following formats:*

```
xx:xx:xx:xx:xx:xx
xxxxxx:xxxxxx
```

dest-mac <*MACaddr*>
Specifies the L2 destination MAC address.

vlan <*vlanID*>
Specifies the name of a VLAN.

in-port-list <*port-list*>
Specifies the SSR ports for which you are setting priority for this flow. The priority applies when the L2 packet enters the SSR on one of the specified ports. The priority does not apply to exit ports.

priority control | high | medium | low | <*trunk-priority*>
Specifies the priority you are assigning to the flow parameters you specified from the list above. You can specify one of the following priorities:

control	Assigns control priority to the IPX flow parameters you have specified. This is the highest priority.
high	Assigns high priority to the IPX flow parameters you have specified.
medium	Assigns medium priority to the IPX flow parameters you have specified.
low	Assigns low priority to the IPX flow parameters you have specified. This is the default.
<*trunk-priority*>	Assigns n 802.1Q VLAN trunk priority when the exit port is a VLAN trunk port. The SSR maps the number to the four internal priorities as follows: 0 = low; 1, 2, or 3 = medium; 4, 5,or 6 = high; 7 = control.

Restrictions

None.

qos set queuing-policy

Purpose

Change the queueing policy from strict priority to weighted fair.

Format

qos set queuing-policy weighted-fair port <*port list*> | **all-ports**

Mode

Configure

Description

The **qos set queuing-policy** command lets you override the default queuing policy (strict priority) in favor of weighted fair queuing on specific ports or on all ports. Only one type of queuing policy can be active at a time.

To set the queuing policy back to strict priority, enter the following command:

```
ssr(config)# no qos set queuing-policy weighted-fair port <port list>
```

Parameters

weighted-fair
Sets the queuing policy to weighted fair.

port <*port list*> | **all-ports**
Specifies the Ethernet ports or WAN modules and ports on which weighted fair queuing apply. Specify **all-ports** to apply weighted fair queuing to all ports.

Restrictions

None.

qos set weighted-fair

Purpose

Set percentages for weighted-fair queuing.

Format

qos set weighted-fair control <*percentage*> **high** <*percentage*> **medium** <*percentage*> **low** <*percentage*> **port** <*port list*> | **all-ports**

Mode

Configure

Description

The **qos set weighted-fair** command lets you set the percentage of SSR bandwidth allocated to the control, high, medium, and low priorities. The percentages apply to specific ports or to all ports. Make sure the total percentages for all four priorities equals 100.

Parameters

control <*percentage*>
Specifies the percentage of SSR bandwidth allocated to the control priority. Specify a number from 1 – 100. The default is 25.

high <*percentage*>
Specifies the percentage of SSR bandwidth allocated to the high priority. Specify a number from 1 – 100. The default is 25.

medium <*percentage*>
Specifies the percentage of SSR bandwidth allocated to the medium priority. Specify a number from 1 – 100. The default is 25.

low <*percentage*>
Specifies the percentage of SSR bandwidth allocated to the low priority. Specify a number from 1 – 100. The default is 25.

port <*port list*> | **all-ports**
Specifies the Ethernet ports or WAN modules and ports on which the defined percentages apply. Specify **all-ports** to apply the percentages to all ports.

Restrictions

The total percentages for all four QoS levels must equal 100%.

qos show ip

Purpose

Show QoS information for IP flows.

Format

qos show ip

Mode

Enable

Description

The **qos show ip** command lets you display QoS information for IP flows.

Parameters

None.

Restrictions

None.

qos show ipx

Purpose

Show QoS information for IPX flows.

Format

qos show ipx

Mode

Enable

Description

The **qos show ipx** command lets you display QoS information for IPX flows.

Parameters

None.

Restrictions

None.

qos show l2

Purpose

Show QoS information for L2 flows.

Format

qos show l2 all-destination all-flow ports <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

Mode

Enable

Description

The **qos show l2** command lets you display QoS information for L2 flows. You can filter the display according to the following:

- Destinations
- Flows
- Ports
- VLANs
- Source MAC addresses
- Destination MAC addresses
- Priority

Parameters

all-destination
Filters the display to show all the L2 destination priorities.

all-flow
Filters the display to show all the L2 flow priorities.

ports <*port-list*>
Filters the display to show L2 priority information for specific ports.

vlan <*vlanID*>
Filters the display to show L2 priority information for specific VLANs.

source-mac <*MACaddr*>Filters the display to show L2 priority information for specific source MAC addresses.

dest-mac <*MACaddr*>
Filters the display to show L2 priority information for specific destination MAC addresses.

Restrictions

None.

qos show

Purpose

Show QoS information for L2, IP, and IPX flows.

Format

qos show ip | ipx | l2 all-destination all-flow ports <*port-list*> **vlan** <*vlanID*> **source-mac** <*MACaddr*> **dest-mac** <*MACaddr*>

Mode

User or Enable

Description

The **qos show** command lets you display QoS information for IP, IPX, and L2 flows. The command shows information for all IP and IPX flows. For L2 flows, you can filter the display according to the following:

- Destinations
- Flows
- Ports
- VLANs
- Source MAC addresses
- Destination MAC addresses
- Priority

Parameters

all-destination
Filters the display to show all the L2 destination priorities.

all-flow
Filters the display to show all the L2 flow priorities.

ports <*port-list*>
Filters the display to show L2 priority information for specific ports.

vlan <*vlanID*>
Filters the display to show L2 priority information for specific VLANs.

source-mac <*MACaddr*>
Filters the display to show L2 priority information for specific source MAC addresses.

dest-mac <*MACaddr*>
Filters the display to show L2 priority information for specific destination MAC addresses.

Restrictions

None.

Chapter 42

radius Commands

The **radius** commands let you secure access to the SSR using the Remote Authentication Dial-In User Service (RADIUS) protocol. When a user logs in to the SSR or tries to access Enable mode, he or she is prompted for a password. If RADIUS authentication is enabled on the SSR, it will contact a RADIUS server to verify the user. If the user is verified, he or she is granted access to the SSR.

Command Summary

Table 30 lists the **radius** commands. The sections following the table describe the command syntax.

Table 30. radius commands

radius accounting command level <*level*>
radius accounting shell start | stop | all
radius accounting snmp active | startup
radius accounting system fatal | error | warning | info
radius authentication login | enable
radius enable
radius set server <*IPaddr*> **radius set [timeout** <*number*>**] [key** <*string*>**] [last-resort password | succeed]**
radius show stats | all

radius accounting command level

Purpose

Causes the specified types of commands to be logged to the RADIUS server.

Format

radius accounting command level <*level*>

Mode

Configure

Description

The **radius accounting command level** command allows you specify the types of commands that are logged to the RADIUS server. The user ID and timestamp are also logged.

Parameters

<level> Specifies the type(s) of commands that are logged to the RADIUS server. Enter one of the following values:

- 5 Log Configure commands.
- 10 Log all Configure and Enable commands.
- 15 Log all Configure, Enable, and User commands.

Restrictions

None.

Example

To cause Configure, Enable, and User mode commands to be logged on the RADIUS server:

```
ssr(config)# radius accounting command level 15
```

radius accounting shell

Purpose

Causes an entry to be logged on the RADIUS server when a shell is stopped or started on the SSR.

Format

radius accounting shell start | stop | all

Mode

Configure

Description

The **radius accounting shell** command allows you to track shell usage on the SSR. It causes an entry to be logged on the RADIUS server when a shell is started or stopped. You can specify that an entry be logged when a shell is started, when a shell is stopped, or when a shell is either started or stopped.

Parameters

start Logs an entry when a shell is started.

stop Logs an entry when a shell is stopped

all Logs an entry when a shell is either started or stopped

Restrictions

None.

Example

To cause an entry to be logged on the RADIUS server when a shell is either started or stopped on the SSR:

```
radius accounting shell all
```

radius accounting snmp

Purpose

Logs to the RADIUS server any changes made to the startup or active configuration via SNMP.

Format

radius accounting snmp active | startup

Mode

Configure

Description

The **radius accounting snmp** command allows you to track changes made to the active or startup configuration through SNMP. It causes an entry to be logged on the RADIUS server whenever a change is made to the ACL configuration. You can specify that an entry be logged to the active or startup configuration.

Parameters

active Logs an entry when a change is made to the active configuration.

startup Logs an entry when a change is made to the startup configuration.

Restrictions

None.

Example

To cause an entry to be logged on the RADIUS server whenever an ACL configuration change is made via SNMP to the active configuration:

```
ssr(config)# radius accounting snmp active
```

radius accounting system

Purpose

Specifies the type(s) of messages to be logged on the RADIUS server.

Format

radius accounting system fatal | error | warning | info

Mode

Configure

Description

The **radius accounting system** command allows you to specify the types of messages that are logged on the RADIUS server.

Parameters

fatal
Logs only fatal messages.

error
Logs fatal messages and error messages.

warning
Logs fatal messages, error messages, and warning messages.

info
Logs all messages, including informational messages.

Restrictions

None.

Example

To log only fatal and error messages on the RADIUS server:

```
ssr(config)# radius accounting system error
```

radius authentication

Purpose

Causes RADIUS authentication to be performed at either the SSR login prompt or when the user tries to access Enable mode.

Format

radius authentication login | enable

Mode

Configure

Description

The **radius authentication** command allows you to specify when RADIUS authentication is performed: either when a user logs in to the SSR, or tries to access Enable mode.

Parameters

login Authenticates users at the SSR login prompt.

enable Authenticates users when they try to access Enable mode.

Restrictions

None.

Example

To perform RADIUS authentication at the SSR login prompt:

```
radius authentication login
```

radius enable

Purpose

Enables RADIUS authentication on the SSR. RADIUS authentication is disabled by default on the SSR.

Format

radius enable

Mode

Configure

Description

The **radius enable** command causes RADIUS authentication to be activated on the SSR. You set RADIUS-related parameters with the **radius set**, **radius accounting shell**, and **radius authorization** commands, then use the **radius enable** command to activate RADIUS authentication.

Parameters

None.

Restrictions

None.

Example

The following commands set RADIUS-related parameters on the SSR. The commands are then activated with the **radius enable** command:

```
radius set server 207.135.89.15
radius set timeout 30
radius authentication login
radius accounting shell all
radius enable
```

radius set

Purpose

Sets parameters for authenticating the SSR through a RADIUS server.

Format

radius set server <*IPaddr*>

radius set [timeout <*number*>**] [key** <*string*>**] last-resort password | succeed**

Mode

Configure

Description

The **radius set** command allows you to set RADIUS-related parameters on the SSR, including the IP address of the RADIUS server, how long to wait for the RADIUS server to authenticate the user, an encryption key, and what to do if the RADIUS server does not reply by a given time.

Parameters

host <*IPaddr*> Is the IP address of a RADIUS server. You can enter up to five RADIUS servers. Enter one server per **radius set server** command.

timeout <*number*> Is the maximum time (in seconds) to wait for a RADIUS server to reply. The default is 3 seconds.

key <*string*> Is an encryption key to be shared with the RADIUS server.

last-resort Is the action to take if a RADIUS server does not reply within the time specified by the **timeout** parameter. If this parameter is *not* specified, user authentication will always fail if the RADIUS server does not reply within the specified timeout period.

Specify one of the following keywords:

password The user is prompted for the password set with **system set password** command. This keyword is *recommended*

for optimal security, however, note that you must set a password with the **system set password** command.

succeed Access to the SSR is granted.

Restrictions

None.

Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are RADIUS servers, and the SSR should wait no more than 30 seconds for a response from one of these servers. If a response from a RADIUS server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the SSR **system set password** command.

```
radius set server 137.72.5.9
radius set server 137.72.5.41
radius set timeout 30
radius set last-resort password
```

radius show

Purpose

Displays information about RADIUS configuration on the SSR.

Format

radius show stats | all

Mode

Enable

Description

The **radius show** command displays statistics and configuration parameters related to RADIUS configuration on the SSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

Parameters

stats Displays the accepts, rejects, and timeouts for each RADIUS server.

all Displays the configuration parameters set with the **radius set** command, in addition to the accepts, rejects, and timeouts for each RADIUS server.

Restrictions

None.

Example

To display configuration parameters and RADIUS server statistics:

```
radius show all
```

Chapter 43

rarpd Commands

The **rarpd** commands let you configure and display information about Reverse Address Resolution Protocol (RARP) on the SSR.

Command Summary

Table 31 lists the **rarpd** commands. The sections following the table describe the command syntax.

Table 31. rarpd commands

rarpd add hardware-address <*mac-address*> **ip-address** <*IPaddr*>
rarpd set interface <*name*> **| all**
**rarpd show interface	mappings**

rarpd add

Purpose

Maps a MAC address to an IP address.

Format

rarp add hardware-address <*mac-address*> **ip-address** <*IPaddr*>

Mode

Configure

Description

The **rarpd add** command allows you to map a MAC address to an IP address for use with RARP. When a host makes a RARP request on the SSR, and its MAC address has been mapped to an IP address with the **rarp add** command, the RARP server on the SSR responds with the IP address that corresponds to the host's MAC address.

Parameters

hardware-address <*mac-address*>
Is a MAC address in the form *xx*:*xx*:*xx*:*xx*:*xx*:*xx* or *xxxxxx*:*xxxxxx*.

ip-address <*IPaddr*>
Is the IP address to be mapped to the MAC address.

Restrictions

None

Example

To map MAC address 00:C0:4F:65:18:E0 to IP address 10.10.10.10:

```
ssr(config)# rarpd add hardware-address 00:C0:4F:65:18:E0 ip-address 10.10.10.10
```

rarpd set interface

Purpose

Specifies the interface(s) to which the SSR's RARP server responds.

Format

rarpd set interface <*name*> | **all**

Mode

Configure

Description

The **rarpd set interface** command allows you to specify which interfaces the SSR's RARP server responds to when sent RARP requests. You can specify individual interfaces or all interfaces.

Parameters

<*name*>	Is the name of an interface.
all	Causes the RARP server to respond to RARP requests from all interfaces.

Restrictions

None.

Example

To cause the SSR's RARP server to respond to RARP requests from interface int1:

```
ssr(config)# rarpd set interface int1
```

rarpd show

Purpose

Displays information about the SSR's RARP configuration.

Format

rarpd show interface | mappings

Mode

Enable

Description

The **rarpd show** command displays information about the configuration of the SSR's RARP server. You can list the MAC-to-IP address mappings or the interfaces to which the SSR responds to RARP requests.

Parameters

interface Lists the interfaces to which the SSR responds to RARP requests.

mappings Displays the list of MAC-to-IP address mappings that was set with the **rarp add command**.

Restrictions

None.

Example

To display the RARP server's list of MAC-to-IP address mappings:

```
ssr(config)# rarpd show mappings
```

Chapter 44

rate-limit Command

The **rate-limit** commands allow you to define rate limits and apply them to IP interfaces.

Command Summary

Table 32 lists the **rate-limit** commands. The sections following the table describe the command syntax.

Table 32. rate-limit commands

rate-limit <*name*> **apply interface** <*interface*>	**all**
rate-limit <*name*> **input acl** <*acl list*> **rate** <*number*> **exceed-action** <*action*> [**sequence** <*number*>]
rate-limit show all | **policy-name** <*name*> | **interface** <*interface*>

rate-limit apply

Purpose

Applies a rate limit definition to an interface.

Format

rate-limit <*name*> **apply inteface** <*interface*> | **all**

Mode

Configure

Description

The **rate-limit apply** command allows you to apply a previously-defined rate limit to an interface.

Parameters

<name>
The name of the rate limit.

interface <*interface*> | **all**
The name of the IP interface. The keyword **all** applies the policy to all IP interfaces.

Restrictions

None.

Examples

To apply a rate limit definition to an interface:

```
ssr(config)# rate-limit client1 apply interface ip16
```

rate-limit input

Purpose

Defines a policy to enable rate limit.

Format

rate-limit <*name*> **input acl** <*acl list*> **rate** <*number*> **exceed-action** <*action*> [**sequence** <*number*>]

Mode

Configure

Description

The **rate-limit input** command allows you to specify the profile for rate limiting by specifying IP ACLs, the rate limit, and the action to be performed if the rate limit is reached. You then use the **rate-limit apply** command to apply the rate limit to an IP interface.

Parameters

<name>
The name of the rate limit.

input acl <*acl list*>
The ACL(s) that define a policy to enable the rate limit. The **rate-limit input** command disregards the **permit/deny** keywords in the ACL rule definition, however, it does look at all parameters in the ACL rule.

rate <*number*>
The rate limit, in bps, for the flow. This value can be between 1 and 1000000000.

exceed-action <*action*>
The action to be taken if the rate limit is reached. Specify one of the following keywords:

drop-packets	Drop the packets.
set-priority-low	Set the priority to low.

set-priority-medium Set the priority to medium.

set-priority-high Set the priority to high.

sequence <*number*>
The sequence number for this policy. This value can be between 1 and 65535.

Restrictions

None.

Examples

To define a rate limit profile 'client1' for the ACL '100' that causes packets to be dropped if the rate limit of 10 million bps is exceeded:

```
ssr(config)# rate-limit client1 input acl 100 rate-limit 10000000
    exceed-action drop-packets
```

rate-limit show

Purpose

Shows rate limit policies.

Format

rate-limit show all | policy-name <*name*> **| interface** <*interface*>

Mode

Enable

Description

The **rate-limit show** command shows information about rate limit policies.

Parameters

all
Displays information on all rate limit policies configured on the SSR.

policy-name <*name*> | **all**
The name of the rate limit. The keyword **all** shows all rate limit policies.

interface <*interface*> | **all**
The name of the IP interface. The keyword **all** shows rate limit policies for all IP interfaces.

Restrictions

None.

Example

To show all configured rate limit policies:

```
ssr# rate-limit show all

------------------------------------------------------------------------
 Rate Limit Policy name : rlpol ❶
 Applied Interfaces : if0 ❷

  ❸           ❹                ❺             ❻         ❼         ❽    ❾
 ACL      Source IP/Mask    Dest. IP/Mask    SrcPort   DstPort   TOS  Prot
 ---      --------------    -------------    --------- --------- ---- ---
 100      10.212.10.11/32   anywhere         any       any       any  IP
 200      10.212.10.12/32   anywhere         any       any       any  IP
 300      10.212.10.13/32   anywhere         any       any       any  IP
 400      10.212.10.14/32   anywhere         any       any       any  IP
 500      10.212.10.10/32   anywhere         any       any       any  IP

   ❿  ⓫          ⓬            ⓭
  Seq ACL      Rate Limit Exceed Action
  --- ----     --------   -------------
  10  100      26000      Low
  10  200      26000      Low
  10  300      26000      Low
  10  400      26000      Low
  10  500      26000      Low
```

Legend:

1. The name of the rate limit.
2. The IP interface to which the rate limit is applied.
3. The name of the ACL(s) that define the rate limit.
4. The source address and filtering mask specified by the ACL.
5. The destination address and filtering mask specified by the ACL.
6. The number of the TCP or UDP source port.
7. The number of the TCP or UDP destination port.
8. The Type of Service value.
9. The protocol for the ACL.
10. The sequence number for this policy.
11. The name of the ACL.
12. The rate limit for the flow.

13. The action to be taken if the rate limit is reached: packets can be dropped or the priority set to low, medium, or high.

Chapter 45

rdisc Commands

The **rdisc** commands allow you to configure router advertisement on the SSR.

Command Summary

Table 33 lists the **rdisc** commands. The sections following the table describe the command syntax.

Table 33. rdisc commands

rdisc add address <*hostname-or-ipaddr*>
rdisc add interface <*name*> **| all**
rdisc set address <*ipaddr*> **type multicast	broadcast advertise enable	disable preference** <*number*> **| ineligible**
rdisc set interface <*name*> **| all min-adv-interval** <*number*> **max-adv-interval** <*number*> **lifetime** <*number*>
rdisc show
rdisc start
rdisc stop

rdisc add address

Purpose

Defines the IP address(es) that are to be included in router advertisements send by the SSR.

Format

rdisc add address <*hostname-or-ipaddr*>

Mode

Configure

Description

The **rdisc add address** command lets you define addresses to be included in router advertisements. If you configure this command, only the specified hostname(s) or IP address(es) are included in the router advertisements.

Parameters

<*hostname-or-ipaddr*>
Defines the hostname or IP address(es) to be included in the router advertisements.

Restrictions

None.

Example

To define an address to be included in router advertisements:

```
ssr(config)# rdisc add address 10.10.5.254
```

rdisc add interface

Purpose

Enables router advertisement on an interface.

Format

rdisc add interface <*name*> | **all**

Mode

Configure

Description

The **rdisc add interface** command lets you enable router advertisement on an interface. By default, all addresses on the interface are included in router advertisements sent by the SSR. If you want to have only specific addresses included in router advertisements, use the **rdisc add address** command to specify those addresses.

Parameters

<*name*> | **all**
The interface on which router advertisement is to be enabled. If **all** is specified, then router advertisement is enabled on all interfaces. By default, router advertisement is disabled on all interfaces.

Restrictions

None.

Example

To enable router advertisement on an interface:

```
ssr(config)# rdisc add interface ssr4
```

rdisc set address

Purpose

Configures router advertisement parameters that apply to a specific address.

Format

rdisc set address <*ipaddr*> **type multicast | broadcast advertise enable | disable preference** <*number*> **| ineligible**

Mode

Configure

Description

The **rdisc set address** command lets you specify the type of router advertisement in which the address is included and the preference of the address for use as a default route.

Parameters

<*ipaddr*>
Specifies the IP address.

type multicast | broadcast
Specifies the type of router advertisement in which the IP address is to be included:

multicast Specifies that the IP address should only be included in a multicast router advertisement. This is the default.

broadcast Specifies that the IP address should only be included in a broadcast router advertisement, even if IP multicast is available.

advertise enable | disable
Specifies whether the IP address is included in the router advertisements:

enable Include the IP address in router advertisements. This is the default.

disable Do not include the IP address in router advertisements.

preference <*number*> **| ineligible**
Specifies the degree of preference of the IP address as a default route. The higher the

value, the more preference. If the IP address is ineligible to be a default route, specify **ineligible**. The default value is 0.

Restrictions

None

Examples

To specify that an address be included only in broadcast router advertisements and that the address is ineligible to be a default route:

```
ssr#(config) rdisc set address 10.20.36.0 type broadcast preference
ineligible
```

rdisc set interface

Purpose

Configures router advertisement parameters that apply to a specific interface or to all interfaces.

Format

rdisc set interface <*name*> | **all min-adv-interval** <*number*> **max-adv-interval** <*number*> **lifetime** <*number*>

Mode

Configure

Description

The **rdisc set interface** command lets you specify the intervals between the sending of router advertisements and the lifetime of addresses sent in a router advertisement.

Parameters

<*name*>
Specifies the name of the interface. If **all** is specified, then the parameters set apply to all interfaces.

min-adv-interval <*number*>
Specifies the minimum time, in seconds, allowed between the sending of unsolicited broadcast or multicast router advertisements. This value can be between 3-1800. The default is 0.75 times the **max-adv-interval** value.

max-adv-interval <*number*>
Specifies the maximum time, in seconds, allowed between the sending of unsolicited broadcast or multicast router advertisements. This value can be between 4-1800. The default value is 600 seconds.

lifetime <*number*>
Specifies the lifetime, in seconds, of addresses in a router advertisement. This value can be between 4-9000. The default is 3 times the **max-adv-interval** value.

Restrictions

None

Examples

To specify the maximum time between the sending of router advertisements on an interface:

```
ssr#(config) rdisc set interface ssr4 max-adv-interval 1200
```

Note that since the **min-adv-interval** and **lifetime** parameters were not specified, the default values for those parameters become 900 seconds and 3600 seconds, respectively.

rdisc show

Purpose

Shows the state of router discovery on the SSR.

Format

rdisc show all

Mode

Enable

Description

The **rdisc show** command shows the state of router discovery on the SSR.

Parameters

all
Displays all router discovery information.

Restrictions

None.

Examples

To display router discovery information:

```
ssr# rdisc show all


Task State: <Foreground NoResolv NoDetach> ❶

    Send buffer size 2048 at 812C68F8
    Recv buffer size 2048 at 812C60D0

Timers:

    RouterDiscoveryServer Priority 30

        RouterDiscoveryServer_SSR2_SSR3_IP <OneShot>
            last: 10:17:21 next: 10:25:05  ❷

Task RouterDiscoveryServer:
    Interfaces:
        Interface SSR2_SSR3_IP: ❸
            Group 224.0.0.1: ❹
                minadvint 7:30 maxadvint 10:00 lifetime 30:00 ❺

                Address 10.10.5.254: Preference: 0 ❻

    Interface policy:
        Interface SSR2_SSR3_IP* MaxAdvInt 10:00 ❼
```

Legend:

1. Information about the RDISC task.
2. Shows when the last router advertisement was sent and when the next advertisement will be sent.
3. The interface on which router advertisement is enabled.
4. Multicast address.
5. Current values for the intervals between the sending of router advertisements and the lifetime of addresses sent in a router advertisement.
6. IP address that is included in router advertisement. The preference of this address as a default route is 0, the default value.
7. Shows configured values for the specified interface.

rdisc start

Purpose

Starts router discovery on the SSR.

Format

rdisc start

Mode

Configure

Description

The **rdisc start** command lets you start router discovery on the SSR. When router discovery is started, the SSR multicasts or broadcasts periodic router advertisements on each configured interface. The router advertisements contain a list of addresses on a given interface and the preference of each address for use as the default route on the interface. By default, router discovery is disabled.

Parameters

None.

Restrictions

None

rdisc stop

Purpose

Stops router discovery.

Format

rdisc stop

Mode

Configure

Description

The **rdisc stop** command stops router discovery on the SSR, thereby stopping router advertisements from being sent out.

Parameters

None.

Restrictions

None

Chapter 46

reboot Command

The **reboot** command reboots the SSR.

Format

reboot

Mode

Enable.

Parameters

None.

Restrictions

None.

Chapter 47

rip Commands

The Routing Information Protocol, Version 1 and Version 2, (RIPv1 and RIPv2) is the most commonly used interior gateway protocol. RIP selects the route with the lowest metric as the best route. The metric is a hop count representing the number of gateways through which data must pass in order to reach its destination. The longest path that RIP accepts is 15 hops. If the metric is greater than 15, a destination is considered unreachable and the SSR discards the route. RIP assumes that the best route is the one that uses the fewest gateways, that is, the shortest path. RIPv1 is described in RFC 1058 and RIPv2 is described in RFC 1723.

Command Summary

Table 34 lists the **rip** commands. The sections following the table describe the command syntax.

Table 34. rip commands

rip add interface | source-gateways | trusted-gateways <*hostname-or-IPaddr*>
rip set auto-summary disable | enable
rip set broadcast-state always | choose | never
rip set check-zero disable | enable
rip set check-zero-metric disable | enable
rip set default-metric <*num*>
rip set interface <*interfacename-or-IPaddr*> **| all [receive-rip enable | disable] [send-rip enable | disable] [metric-in** <*num*>**] [metric-out** <*num*>**] [version 1 | version 2 [type broadcast | multicast]] authentication-method [none | (simple | md5 key-chain** <*num-or-string*>**)]**

Table 34. rip commands (Continued)

rip set poison-reverse disable | enable
rip set preference <*num*>
rip show <*option-list*>
rip start
rip stop
rip trace [packets | request | response | local-options] [detail] [send | receive]

rip add

Purpose

Adds RIP entities.

Note: By default, RIP is disabled on all SSR interfaces. To enable RIP on an interface, you must use the **rip add interface** command.

Format

rip add interface <*interfacename-or-IPaddr*>

rip add source-gateways | trusted-gateways <*hostname-or-IPaddr*>

Mode

Configure

Description

The **rip add** command lets you add the following RIP entities:

- Interfaces that will run RIP
- Routers that send RIP updates directly, rather than through broadcast or multicast
- Trusted gateways, from which the SSR will accept RIP updates. when you add trusted gateways, the SSR does not accept RIP updates from sources other than those trusted gateways.

Parameters

interface
Informs the RIP process about the specified interfaces. You can specify a list of interface names or IP addresses or use the **all** keyword to specify all interfaces.

source-gateways
Adds a router that sends RIP updates directly, rather than using broadcasts or multicasts. You can specify a single interface name or IP address.

Note: Updates to source gateways are not affected by the RIP packet transmission state of the interface.

trusted-gateway
Adds a trusted source for RIP updates. When you add trusted gateways, the SSR will not accept RIP updates from any sources except the trusted gateways. You can specify a single interface name or IP address.

<interfacename-or-IPaddr>
The interface name or IP address of the interface, router, or gateway. You can specify a list or use the keyword **all** to specify all SSR interfaces.

<hostname-or-IPaddr>
The hostname or IP address of the source or trusted gateway.

Restrictions

None.

rip set auto-summary

Purpose

Enables automatic summarization and redistribution of RIP routes.

Format

rip set auto-summary disable | enable

Mode

Configure

Description

The **rip set auto-summary** command specifies that routes to subnets should be automatically summarized by the classful network boundary and redistributed into RIP.

Parameters

disable | enable
Enables or disables automatic summarization and redistribution of RIP routes.

Restrictions

None.

rip set broadcast-state

Purpose

Determines if RIP packets will be broadcast regardless of the number of interfaces present. This is useful when propagating static routes or routes learned from another protocol into RIP. In some cases, the use of broadcast when only one network interface is present can cause data packets to traverse a single network twice.

Format

rip set broadcast-state always | choose | never

Mode

Configure

Description

The **rip set broadcast-state** command specifies whether the SSR broadcasts RIP packets regardless of the number of interfaces present.

Parameters

always | choose | never
Specifies whether the SSR broadcasts RIP packets regardless of the number of interfaces present. Specify one of the following:

always Always sends RIP broadcasts regardless of the number of interfaces present.

choose Sends RIP broadcasts only if more than one interface is configured on the SSR. This is the default state.

never Never sends RIP broadcasts on attached interfaces.

Restrictions

None.

rip set check-zero

Purpose

Specifies whether RIP should make sure that reserved fields in incoming RIP V1 packets are zero. Normally, RIP will reject packets where the reserved fields are non-zero.

Format

rip set check-zero disable | enable

Mode

Configure

Description

The **rip set check-zero** command specifies whether RIP should make sure that reserved fields in incoming RIP V1 packets are zero. RIP will reject packets where the reserved fields are non-zero.

- If you use the **disable** keyword, RIP does not check the reserved field.
- If you use the **enable** keyword, RIP on the SSR checks to ensure that the reserved fields in incoming RIP packets are zero. If the reserved field in a RIP packet is not zero, the SSR discards the packet. This is the default state.

Parameters

disable | enable
Enables or disables checking of the reserved field.

Restrictions

None.

rip set check-zero-metric

Purpose

Specifies whether RIP should accept routes with a metric of zero. Normally, RIP will reject routes with a metric of zero.

Format

rip set check-zero-metric disable | enable

Mode

Configure

Description

The **rip set check-zero-metric** command specifies whether RIP should accept routes with a metric of zero. This may be necessary for interoperability with other RIP implementations that send routes with a metric of zero.

- If you use the **disable** keyword, RIP accepts routes that have a metric of zero and treats them as though they were received with a metric of 1.
- If you use the **enable** keyword, RIP rejects routes that have a metric of zero. This is the default state.

Parameters

disable | enable
Enables or disables acceptance of RIP routes that have a metric of zero.

Restrictions

None.

rip set default-metric

Purpose

Defines the metric used when advertising routes via RIP that were learned from other protocols. If not specified, the default value is 16 (unreachable). This choice of values requires you to explicitly specify a metric in order to export routes from other protocols into RIP. This metric may be overridden by a metric specified in the export command.

Note: The metric 16 is equivalent in RIP to "infinite" and makes a route unreachable. You must set the default metric to a value other than 16 in order to allow the SSR to export routes from other protocols such as OSPF and BGP-4 into RIP.

Format

rip set default-metric <*num*>

Mode

Configure

Description

The **rip set default metric** command defines the metric used when advertising routes via RIP that were learned from other protocols.

Parameters

<*num*> Specifies the metric. Specify a number from 1 – 16. The default is 16.

Restrictions

None.

rip set interface

Purpose

Set the RIP state, version, type of update messages, metric and authentication scheme used for each interface running RIP.

Format

rip set interface <*interfacename-or-IPaddr*> **| all**

[advertise-classfull enable | disable]

[receive-rip enable | disable]

[send-rip enable | disable]

[metric-in <*num*>**]**

[metric-out <*num*>**]**

[version 1 | version 2 [type broadcast | multicast]]

[authentication-method none | (simple | md5
key-chain <*num-or-string*>**)]**

Mode

Configure

Description

The **rip set interface** command lets you set the following parameters for RIP interfaces:

- Whether the interface will accept RIP updates
- Whether the interface will send RIP updates
- The RIP version (RIP V1 or RIP V2)
- The packet type used for RIP V2 updates (broadcast or multicast)
- The metric added to incoming RIP updates
- The metric added to outgoing RIP updates

- The key-chain for RIP update authentication
- The authentication method used for RIP updates (none, simple, or MD5)

Parameters

<interfacename-or-IPaddr> **| all**
The interface names or IP addresses of the interfaces for which you are setting RIP parameters. Specify the **all** keyword if you want to set RIP parameters for all IP interfaces on the SSR.

advertise-classfull enable | disable
This command is used to announce a classfull network onto a subnetted RIP Version 1 interface having the same classfull network.

receive-rip enable | disable
Specifies whether the interface(s) can receive RIP updates. Specify **enable** if you want to receive RIP updates on the interface. Otherwise, select **disable**.

The default is **enable**.

Note: This option affects RIP updates sent from trusted gateways. If you specify **disable**, the SSR will not receive any RIP updates, including those sent from trusted gateways. If you specify **enable** *and* you have set up trusted gateways, the SSR will accept updates only from those trusted gateways.

send-rip enable | disable
Specifies whether the interface(s) can send RIP updates. Specify **enable** if you want to send RIP updates from this interface. Otherwise, specify **disable**.

The default is **enable**.

Note: This option does not affect the sending of updates to source gateways.

metric-in *<num>*
Specifies a metric that the interface adds to incoming RIP routes before adding them to the interface table. Specify a metric from 1 – 16. Use this option to make the SSR prefer RIP routes learned from the specified interfaces less than RIP routes from other interfaces. The default is 1.

metric-out *<num>*
Specifies a metric that the interface adds to outgoing RIP routes sent through the specified interfaces. The default is 0. Use this option to make other routers prefer other sources of RIP routes over this router.

version 1 | version 2 [type broadcast | multicast]
Specifies the RIP version used on the interface(s).

broadcast
Causes RIP V2 packets that are RIP V1-compatible to be broadcast on this interface.

multicast
Causes RIP V2 packets to be multicasted on this interface; this is the default.

authentication-method none | (simple | md5 key-chain <*num-or-string*>**)**
The authentication method the interface uses to authenticate RIP updates. Specify one of the following:

none
The interface does not use any authentication.

simple
The interface uses a simple password in which an authentication key of up to 8 characters is included in the packet.

md5
The interface uses MD5 authentication. This method uses the MD5 algorithm to create a crypto-checksum of a RIP packet and an authentication key of up to 16 characters.

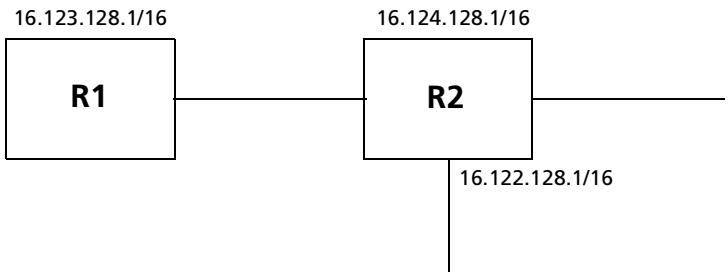
Note: If you choose the simple or md5 authentication method, you must also specify a key-chain identifier using the key-chain option.

key-chain <*num-or-string*>
The identifier of the key-chain containing the authentication keys. This parameter applies only if you specified simple or md5 for the authentication type.

Restrictions

None.

Example



In this example, router R1 has the following three interfaces:

1. It is connected to router R2 over interface 16.123.128.1/16. It is running RIP version 1 on this interface.
2. It has two other interfaces with the following addresses (16.124.128.1/16, 16.122.128.1/16).

3. Router R1 the entire class A network (16.0.0.0/8) behind it.

By default, router R1 would not announce a classful network (16.0.0.0/8) over a subnet (16.123.128.1/16). If that is something which is desired, then the below given command should be entered.

```
rip set interface 16.123.128.1 advertise-classfull enable | disable
```

Typically, a user would enable automatic summarization for RIP. This would create an implicit aggregate 16.0.0.0/8. If it is desired, that this classfull network is announced over a subnetted RIP Version 1 interface, then the above command should be entered.

rip set poison-reverse

Purpose

Enables poison reverse on all SSR interfaces.

Format

rip set poison-reverse disable | enable

Mode

Configure

Description

The **rip set poison-reverse** command allows you to enable or disable poison reverse on all SSR interfaces. The SSR supports poison reverse as specified by RFC 1058.

Note: Turning on poison reverse will approximately double the amount of RIP updates.

Parameters

disable | enable
Enables or disables poison reverse on the SSR.

Restrictions

None.

rip set preference

Purpose

Sets the preference of routes learned from RIP. The default preference is 100. This preference may be overridden by a preference specified in the import command.

Format

rip set preference <*num*>

Mode

Configure

Description

The **rip set preference** command sets the preference for destinations learned through RIP. The preference you specify applies to all IP interfaces for which RIP is enabled on the SSR. The default preference is 100. You can override this preference by specifying a different preference in an import policy.

Parameters

<*num*> Specifies the preference. Specify a number from 0 – 255. The default is 100. Lower numbers have higher preference.

Restrictions

None.

rip show

Purpose

Display RIP information.

Format

rip show <*option-list*>

Mode

Enable

Description

The **rip show** command displays RIP information.

Parameters

<*option-list*>
Specifies the RIP dump information you want to display. Specify one or more of the following:

all
Displays all RIP tables.

globals
Displays RIP globals.

timers
Displays RIP timers.

interface
Displays RIP interfaces.

active-gateways
Displays active gateways running RIP.

interface-policies
Displays RIP interface policies.

import-policies
Displays RIP import policies.

export-policies
Displays RIP export policies.

Restrictions

None.

rip start

Purpose

Start RIP on the SSR.

Note: RIP is disabled by default.

Format

rip start

Mode

Configure

Description

The **rip start** command starts RIP on all IP interfaces on the SSR for which RIP is enabled.

Parameters

None.

Restrictions

None.

rip stop

Purpose

Stop RIP on the SSR.

Format

rip stop

Mode

Configure

Description

The **rip stop** command stops RIP on all IP interfaces on the SSR for which RIP is enabled.

Parameters

None.

Restrictions

None.

rip trace

Purpose

Trace RIP packets.

Format

rip trace [packets | request | response | local-options] [detail | send | receive]

Mode

Configure

Description

The **rip trace** command traces the following sets of RIP packets:

- RIP request packets sent or received by the SSR
- RIP response packets sent or received by the SSR

Depending on the options you specify, you can trace all packets, request packets only, or receive packets only. In addition, you can select to trace the request packets, receive packets, or both that are sent by the SSR, received by the SSR, or all packets (both sent packets and received packets).

Parameters

packets Traces all RIP packets, both request packets and response packets. This is the default.

request Traces only request packets, such as REQUEST, POLL and POLLENTRY packets.

response Traces only response packets.

For the **packets**, **request**, and **response** parameters, you can optionally specify one of the following:

detail Shows detailed information about the traced packets.

receive Shows information about traced RIP packets received by the SSR.

send Shows information about traced RIP packets sent by the SSR.

Note: The default is to show both send and receive packets.

local-options Sets trace options for this protocol only. These trace options are inherited from those set by the **ip-router global set trace options** command, or you can override them here. Specify one or more of the following:

all Turns on all tracing.

general Turns on normal and route tracing.

state Traces state machine transitions in the protocols.

normal Traces normal protocol occurrences.

Note: Abnormal protocol occurrences are always traced.

policy Traces application of protocol and user-specified policies to routes being imported and exported.

task Traces system processing associated with this protocol or peer.

timer Traces timer usage by this protocol or peer.

route Traces routing table changes for routes installed by this protocol or peer.

Restrictions

None.

Chapter 48

rmon Commands

The **rmon** commands let you display and set parameters for RMON statistics on a per-port basis. RMON information corresponds to RFCs 1757 and 2021.

Command Summary

Table 35 lists the **rmon** commands. The sections following the table describe the command syntax.

Table 35. rmon commands

rmon address-map index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable** | **disable**]
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**]
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**]
rmon apply cli-filters <*filter id*>
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**]
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**]

Table 35. rmon commands (Continued)

rmon clear cli-filter
rmon enable
rmon etherstats index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable	disable**]
rmon event index <*index-number*> **type none	log	trap	both** [**community** <*string*>] [**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**]
rmon history index <*index-number*> **port** <*port*> [**interval** <*seconds*>] [**owner** <*string*>] [**samples** <*num*>] [**status enable	disable**]
rmon hl-host index <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable	disable**]
rmon hl-matrix index <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable	disable**]
rmon host index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable	disable**]
rmon host-top-n index <*index-number*> **host-index** <*number*> [**base** <*statistics*>] [**duration** <*time*>] [**size** <*size*>] [**owner** <*string*>] [**status enable	disable**]
rmon matrix index <*index-number*> [**port** <*port*>] [**owner** <*string*>] [**status enable	disable**]
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**]
rmon protocol-distribution index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable	disable**]
**rmon set lite	standard	professional default-tables yes	no**
rmon set cli-filter <*filter-id*> <*parameter*>
rmon set memory <*number*>
rmon set ports <*port list*> **| allports**
rmon set protocol-directory <*protocol*> **| all-protocols [address-map on	off	na] [host on	off	na] [matrix on	off	na]**
rmon show <*rmon-parm* >
rmon user-history-apply <*groupname*> **to** <*user-history-index*>

Table 35. rmon commands (Continued)

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 address-map

Purpose

Configures the RMON 2 Address Map group.

Format

rmon address-map index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The Address Map group maps MAC addresses to network address bindings that are discovered by the SSR on a per-port basis. The **rmon address-map** command sets various parameters of the RMON 2 Address Map table.

If the default tables were turned on for the Professional group, an entry in the Address Map control table is created for each available port.

Use the **rmon show address-map** command to display the address map.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Address Map table.

<*port*>
Specifies the port from which to collect data.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To create an entry in the Address Map table for port et.1.3:

```
ssr(config)# rmon address-map index 20 port et.1.3
```

rmon al-matrix-top-n

Purpose

Gathers the top *n* Application Layer Matrix entries.

Format

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**]

Mode

Configure

Description

The **rmon al-matrix-top-n** command gathers the top *n* Application Layer Matrix entries sorted by a specified statistic. To do this, you must first configure the Application Layer/Network Layer Matrix table using the **rmon hl-matrix** command.

Use the **rmon show al-matrix-top-n** command to display the top *n* Application Layer Matrix entries.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the application layer matrix table.

matrix-index <*number*>
Specifies the index into the hl-matrix table. The default is 0.

ratebase terminal-packets | terminal-octets | all-packets | all-octets
Specifies the sorting method:

terminal-packets Sort by terminal packets.

terminal-octets Sort by terminal octets.

all-packets Sort by all packets.

all-octets Sort by all octets.

duration <*number*>
Specifies the duration, in seconds, between reports. If the duration is 0 (the default), this implies that no reports have been requested for this entry. The default is 0.

size <*number*>
Specifies the maximum number of matrix entries to include in the report. The default is 150.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To monitor the top *n* entries in the Application Layer Matrix, you should first configure the Application Layer/Network Layer Matrix table using the **rmon hl-matrix** command. Then, to gather the top 100 Application Layer Matrix entries sorted by all packets, use the following command:

```
ssr(config)# rmon al-matrix-top-n index 25 matrix-index 50 ratebase all-
    packets duration 60 size 100
```

rmon alarm

Purpose

Configures the RMON 1 Alarm group.

Format

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**]

Mode

Configure

Description

The Alarm group takes periodic statistical samples and compares them with previously-configured thresholds. If a monitored variable crosses a threshold, an alarm is generated. The **rmon alarm** command sets various parameters of the RMON 1 Alarm control table.

Use the **rmon show alarm** command to display the alarm data.

Parameters

<*index-number*>
Is a number that uniquely identifies an entry in the alarm table. The value must be between 1 and 65535, inclusive.

interval <*seconds*>
Specifies the sampling interval in seconds when statistical samples of variables are collected and compared to the rising and falling thresholds. The value must be between 1 and 2147483647, inclusive.

falling-event-index <*num*>
Is the action to be taken as defined by the row with this index in the event table when a falling threshold is crossed. The value must be between 1 and 65535, inclusive.

falling-threshold <*num*>
Specifies that the sample's value must be less than or equal to the threshold to trigger

an alarm. When the sample's value is less than or equal to this threshold, and the value at the last sampling interval was greater than this threshold, a single event is generated. The value must be between 1 and 2147483647, inclusive.

owner <*string*>
Specifies the owner of the alarm resource; for example, an IP address, machine name or person's name.

rising-event-index <*num*>
Is the action to be taken as defined by the row with this index in the event table when a rising threshold is crossed. The value must be between 1 and 65535, inclusive.

rising-threshold <*num*>
Specifies that the sample's value must be greater than or equal to the threshold to trigger an alarm. When the sample's value is greater than or equal to this threshold, and the value at the last sampling interval was less than this threshold, a single event is generated. The value must be between 1 and 2147483647, inclusive.

startup <*keyword*>
Specifies the condition for which the alarm is to be generated. The condition can be one of the following:

rising	Causes an alarm to be generated if the sampled variable is greater than or equal to the rising threshold.
falling	Causes an alarm to be generated if the sampled variable is less than or equal to the falling threshold.
both	Causes an alarm to be generated if the sampled variable is greater than or equal to the rising threshold or less than or equal to the falling threshold.

status enable | disable
Enables or disables this alarm.

type <*keyword*>
Specifies the method of sampling the selected variable and calculating the value to be compared against the thresholds. The sampling method can be one of the following:

absolute-value	Monitor the absolute value over the sample interval of the variable against the threshold value.
delta-value	Monitor the change in value over the sample interval of the variable against the threshold value.

variable <*string*>
Specifies the object identifier of the variable to be sampled. Only variables that resolve to an ASN.1 primitive type of INTEGER may be sampled.

Restrictions

None.

Examples

To cause an alarm event if the variable defined in alarm 10 crosses the rising threshold:

```
ssr(config)# rmon alarm index 10 startup rising interval 30 variable
    1.3.6.1.2.1.5.14.0 rising-threshold 40 rising-event-index 1
```

To monitor the absolute value of the variable against a threshold value:

```
ssr(config)# rmon alarm index 10 type absolute-value startup rising
    interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40 rising-
    event-index 1
```

To specify Mike as the owner of alarm 10:

```
ssr(config)# rmon alarm index 10 owner Mike type absolute-value startup
    rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40
    rising-event-index 1
```

To specify a 5-second interval on alarm 10:

```
ssr(config)# rmon alarm index 10 interval 5 type absolute-value startup
    rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-threshold 40
    rising-event-index 1
```

To specify the rising threshold at 10 on alarm 10:

```
ssr(config)# rmon alarm index 10 rising-threshold 10 type delta-value
    startup rising interval 30 variable 1.3.6.1.2.1.5.14.0 rising-
    event-index 1
```

rmon apply cli-filters

Purpose

Apply a specific CLI RMON filter.

Format

rmon apply cli-filters <*filter id*>

Mode

Enable

Description

The **rmon apply cli-filters** command applies a specific CLI RMON filter to the current Telnet or Console session. This enables different users to select the different CLI filters which you should define using the **rmon set cli-filter** command.

Use the **rmon show cli-filters** command to see the RMON CLI filters that have been defined on the SSR. Use the **rmon clear cli-filter** command to clear the applied filter.

Parameter

<*filter id*> Is a number between 1 and 65535 that identifies the filter ID to apply.

Restrictions

None.

Example

To apply filter ID 2:

```
ssr> rmon apply cli-filters 2
```

To see a list of CLI RMON filters:

```
ssr> rmon show cli-filters
RMON CLI Filters
Id   Filter
--   ------
 1   (inpkts >= 0)
 2   (inpkts >= 0 and outoctets >= 0)
 3   srcmac 222222222222 and (outoctets >= 0)
You have selected a filter: (inpkts >= 0)
```

rmon capture

Purpose

Configures the RMON 1 Packet Capture group.

Format

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**]

Mode

Configure

Description

The Packet Capture group allows packets to be captured after they have flowed through a channel. The **rmon capture** command sets various parameters of the RMON 1 Packet Capture table.

Use the **rmon show packet-capture** command to display the Packet Capture table.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Packet Capture table.

channel-index <*number*>
Is a number between 1 and 65535 that identifies the channel that is the source of packets. The default is 0.

full-action lock | wrap
Specifies the action of the buffer when it reaches the full status:

lock Stop capturing packets when the buffer reaches the full status.

wrap Wrap around when the buffer reaches the full status.

slice-size <*number*>
Is a number between 0 and 2147483647 that is the maximum number of octets that will be saved in this capture buffer. The default is 100.

download-slice-size <*number*>
Is a number between 0 and 2147483647 that is the maximum number of octets that will be returned in an SNMP retrieval. The default is 100.

download-offset <*number*>
Is a number between 0 and 2147483647 that is the offset of the first octet of each packet that will be returned in an SNMP retrieval. The default is 0.

max-octets <*number*>
Is a number between 0 and 2147483647 that is maximum number of octets to be saved. The default is 1.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this channel. The default is enable.

Restrictions

None.

Example

To create an entry in the Packet Capture table:

```
ssr(config)# rmon capture index 20 channel-index 1 full-action wrap
```

rmon channel

Purpose

Configures the RMON 1 Filter Channel group.

Format

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**]

Mode

Configure

Description

The Filter Channel group must be configured in order to configure the Filter group. The **rmon channel** command sets various parameters of the RMON 1 Filter Channel table. After a channel row has been created, a filter must be defined with the **rmon filter** command.

Use the **rmon show channels** command to display all the channels configured on the SSR.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Filter Channel table.

port <*port*>
Identifies the port from which data is collected.

accept-type matched | failed
Specifies the action of the filters associated with this channel:

matched Packets will be accepted if they are accepted by both the packet data and packet status matches of an associated filter.

failed Packets will be accepted only if they fail either the packet data match or the packet status match of each of the associated filters.

data-control on | off
Specifies the flow control of the data:

on Implies data, status, and events flow through this channel.

off Implies data, status, and events will not flow through this channel.

turn-on-event-index <*number*>
Is a number between 0 and 65535 that identifies the event configured to turn the associated data control from off to on.

turn-off-event-index <*number*>
Is a number between 0 and 65535 that identifies the event configured to turn the associated data control from on to off.

event-index <*number*>
Is a number between 0 and 65535 that identifies the event configured to be generated when the associated data control is on and a packet is matched.

channel-status ready | always-ready
Specifies the status:

ready A single event is generated.

always-ready Allows events to be generated at will.

description <*string*>
Describes this channel in a maximum of 127 bytes.

owner <*string*>
Specifies the owner of packet capture; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this channel. The default is enable.

Restrictions

None.

Example

To create an entry in the Filter Channel table:

```
ssr(config)# rmon channel index 25 port et.1.3 accept-type matched data-
control on turn-on-event-index 30 turn-off-event-index 55 event-index
60 channel-status ready
```

rmon clear cli-filter

Purpose

Clear the currently-selected CLI RMON filter.

Format

rmon clear cli-filter

Mode

Enable

Description

The **rmon clear cli-filter** command clears the CLI RMON filter that was applied with the **rmon apply cli-filters** command.

Parameters

None.

Restrictions

None.

rmon enable

Purpose

Enables RMON.

Format

rmon enable

Mode

Configure

Description

When the SSR is booted, RMON is off by default. The **rmon enable** command turns RMON on. At least one of the Lite, Standard, or Professional RMON groups must be configured first before you can turn on RMON. Use the **rmon set** command to configure the Lite, Standard, or Professional RMON groups.

To disable RMON, the **rmon enable** command must be negated. This frees up all resources associated with RMON, including any memory allocated to RMON.

Parameters

None.

Restrictions

If the SNMP agent is disabled, RMON cannot be enabled. If RMON is enabled and the SNMP agent is disabled, then RMON will be turned off.

rmon etherstats

Purpose

Configures the RMON 1 Ethernet Statistics (Etherstats) group.

Format

rmon etherstats index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The Etherstats group contains statistics for SSR ports. The **rmon etherstats** command sets various parameters of the RMON 1 Etherstats control table. If default tables were turned on for the Lite group, a entry is created in the Etherstats control table for each available port.

Use the **rmon show etherstats** command to display the Etherstats data.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Etherstats control table.

port <*port*>
Specifies the physical port from which to collect data.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this Etherstat. The default is enable.

Restrictions

None.

Example

To create an entry in the Etherstats control table:

```
ssr(config)# rmon etherstats index 10 port et.1.3
```

rmon event

Purpose

Configures the RMON 1 Event group.

Format

rmon event index <*index-number*> **type none | log | trap | both** [**community** <*string*>] [**description** <*string*>] [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The Event group controls the generation and notification of events. The **rmon event** command sets various parameters of the RMON 1 Event control table.

Use the **rmon show event** command to display the event data.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies an entry in the Event table.

community <*string*>
Specifies the SNMP community string to be sent with the trap. If an SNMP trap is to be sent, it will go to the SNMP community specified in this string.

description <*string*>
Specifies a comment describing this event.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this event. The default is enable.

type none | log | trap | both
Specifies what action to be taken when the event occurs. The action can be one of the following:

none Causes no notification to be sent for the event.

log Causes an entry for the event to be made in the log table for each event.

trap Causes an SNMP trap to be sent to one or more management stations for the event.

both Causes both an entry to be made in the log table and an SNMP trap to be sent to one or more management stations.

Restrictions

None.

Examples

To set the event community string to public:

```
ssr(config)# rmon event index 10 community public
```

To add the description “num-pkts” to event 10:

```
ssr(config)# rmon event index 10 description num-pkts
```

To specify Ed as the owner of event 10:

```
ssr(config)# rmon event index 10 owner Ed
```

To send an SNMP trap when event 10 is triggered:

```
ssr(config)# rmon event index 10 type trap
```

rmon filter

Purpose

Configures the RMON 1 Filter group.

Format

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**]

Mode

Configure

Description

The Filter group allows packets to be matched on certain criteria. The **rmon filter** command sets various parameters of the RMON 1 Filter table. To configure the Filter group, the Filter Channel group must first be configured with the **rmon channel** command.

Use the **rmon show filters** command to display the filters defined on the SSR.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Filter table.

channel-index <*number*>
Is a number between 1 and 65535 that identifies the channel of which this filter is a part.

data-offset <*number*>
Is a number between 0 and 2147483647 that is the offset from the beginning of each packet where a match of packet data will be attempted.

data <*string*>
Is a string of up to 512 characters that is the data that is to be matched with the input packet.

data-mask <*string*>
Is a string of up to 512 characters that is the mask that is applied to the match process.

data-not-mask <*string*>
Is a string of up to 512 characters that is the inversion mask that is applied to the match process.

pkt-status <*number*>
Is a number between 0 and 2147483647 that is the status that is to be matched with the input packet.

status-mask <*number*>
Is a number between 0 and 2147483647 that is the mask that is applied to the status match process.

status-not-mask <*number*>
Is a number between 0 and 2147483647 that is the inversion mask that is applied to the status match process.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this channel. The default is enable.

Restrictions

None.

Example

To create an entry in the Filter table:

```
ssr(config)# rmon filter index 25 channel-index 35 data kgreen
```

rmon history

Purpose

Configures the RMON 1 History group.

Format

rmon history index <*index-number*> **port** <*port*> [**interval** <*seconds*>] [**owner** <*string*>] [**samples** <*num*>] [**status enable | disable**]

Mode

Configure

Description

The RMON History group periodically records samples of variables and stores them for later retrieval. You use the r**mon history** command to specify the SSR port to collect data from, the number of samples, the sampling interval, and the owner. If default tables were turned on for the Lite group, an entry is created in the History control table for each available port.

Use the **rmon show history** command to display the history data.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies an entry in the History table.

interval <*seconds*>
Specifies the sampling interval in seconds. This value must be between 1 and 3600, inclusive. The default value is 1800.

owner <*string*>
Specifies the owner of the history resource; for example, an IP address, machine name or person's name.

port <*port*>
Specifies the port from which to collect data.

samples <*num*>
Specifies the number of samples to be collected before wrapping counters. This value must be between 1 and 65535, inclusive. The default value is 50.

status enable | disable
Enables or disables this history control row.

Restrictions

None.

Example

To specify that port et.3.1 collect 60 samples at an interval of 30 seconds:

```
ssr(config)# rmon history index 10 port et.3.1 samples 60 interval 30
```

rmon hl-host

Purpose

Configures the RMON 2 Application Layer and Network Layer Host groups.

Format

rmon hl-host index <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable** | **disable**]

Mode

Configure

Description

The **rmon hl-host** command sets various parameters of the RMON 2 Application Layer and Network Layer Host groups. The Application Layer Host group monitors traffic from the network layer up to the application layer for any protocol communication defined in the protocol directory. The Network Layer Host group monitors traffic at the network layer for any protocol defined in the protocol directory.

Configuration of the Application Layer/Network Layer Host table involves configuring only one control row in the Application Layer Host control table. This table, when configured, captures both application layer and network layer host data. If the default tables were turned on for the Professional group, an entry is created in the Application Layer Host control table for each available port.

Use the **rmon show al-host** command to display the Application Layer Host table. Use the **rmon show nl-host** command to display the Network Layer Host table.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the application layer host control table.

<*port*>
Specifies the port from which to collect data.

nl-max-entries
Specifies the maximum number of network layer entries. The default is 1.

al-max-entries
Specifies the maximum number of application layer entries. The default is 1.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To create an entry in the Application Layer Host control table:

```
ssr(config)# rmon hl-host index 20 port et.1.3
```

rmon hl-matrix

Purpose

Configures the RMON 2 Application Layer Matrix and Network Layer Matrix groups.

Format

rmon hl-matrix index <*index-number*> **port** <*port*> **nl-max-entries** <*number*> **al-max-entries** <*number*> [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The **rmon hl-matrix** command sets various parameters of the RMON 2 Application Layer Matrix and Network Layer Matrix groups. The Application Layer Matrix group monitors traffic from the network layer up to the application layer for any protocol communication defined in the protocol directory. The Network Layer Matrix group monitors traffic at the network layer for any protocol defined in the protocol directory.

Configuration of the Application Layer/Network Layer Matrix table involves configuring only one control row in the Application Layer Matrix control table. When configured, this table captures both application layer and network layer matrix data. If the default tables were turned on for the Professional group, an entry is created in the Application Layer Matrix control table for each available port.

Use the **rmon show al-matrix** command to display the Application Layer Matrix table. Use the **rmon show nl-matrix** command to display the Network Layer Matrix table.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the application layer matrix control table.

<*port*>
Specifies the port from which to collect data.

nl-max-entries <*number*>
Specifies the maximum number of network layer entries. The default is 1.

al-max-entries <*number*>
Specifies the maximum number of application layer entries. The default is 1.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To create an entry in the Application Layer Matrix control table:

```
ssr(config)# rmon hl-matrix index 20 port et.1.3
```

rmon host

Purpose

Configures the RMON 1 Host group.

Format

rmon host index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The RMON 1 Host group captures L2 information from hosts coming in on a particular port. The **rmon host** command sets various parameters of the Host group. If default tables were turned on for the standard group, an entry is created in the Host control table for each available port.

Use the **rmon show hosts** command to display the host data and logs.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Host table.

port <*port*>
Specifies the physical port from which to collect data.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this host. The default is enable.

Restrictions

None.

Example

To create an entry in the Host control table:

```
ssr(config)# rmon hosts index 20 port et.1.3
```

rmon host-top-n

Purpose

Configures the RMON 1 HostTopN group.

Format

rmon host-top-n index <*index-number*> **host-index** <*number*> [**base** <*statistics*>] [**duration** <*time*>] [**size** <*size*>] [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The HostTopN group displays the top n number of hosts, sorted by a specified statistic. The **rmon host-top-n** command sets various parameters of the RMON 1 HostTopN control table. The HostTopN group depends upon the Host group and the host-index specified in the HostTopN control table must correspond to a pre-defined host index in the Host control table.

Use the r**mon show host-top-n** command to display the control table row.

Note that Host Top N report runs once. To run the reports again via the CLI, the control row must be disabled and then enabled. If the report has already been run, the Time Remaining field is set to zero. Otherwise, the Time Remaining field will be decremented until the report is run.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Host Top N table.

<*number*>
Is a number between 1 and 65535 that is the index into the host table identified by hostIndex.

<*statistics*>
Specifies the type of statistic from which to collect data. Specify one of the following keywords:

in-packets Gather top statistics according to In-Packets.

out-packets
Gather top statistics according to Out-Packets.

in-octets Gather top statistics according to In-Octets.

out-octets Gather top statistics according to Out-Octets.

out-errors Gather top statistics according to Out-Errors.

out-broadcastPkts
Gather top statistics according to Out-BroadcastPkts.

out-multicastPkts
Gather top statistics according to Out-MulticastPkts.

<*time*>
Is a number between 1 and 2147483647 that is the duration, in seconds, between reports. The default is 0.

<size>
Is a number between 1 and 2147483647 that is the maximum number of hosts to include in the table. The default is 10.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this hostTopN. The default is enable.

Restrictions

None.

Example

To create an entry in the HostTopN control table:

```
ssr(config)# rmon host-top-n index 25 host-index 55 base in-packets
duration 60 size 24
```

rmon matrix

Purpose

Configures the RMON 1 Matrix group.

Format

rmon matrix index <*index-number*> [**port** <*port*>] [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The Matrix group captures L2 traffic on a particular port between two hosts (a source MAC and destination MAC address). The **rmon matrix** command sets various parameters of the RMON 1 Matrix control table. If default tables were turned on for the Standard group, an entry is created in the Matrix control table for each available port.

Note: By default, ports on the SSR operate in address-bridging mode. The port must be enabled in *flow-bridging* mode in order for layer 2 matrix information to be captured.

Use the **rmon show matrix** command to display the matrix group and logs.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Matrix table.

<*port*>
Specifies the port from which to collect data.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To create an entry in the Matrix control table:

```
ssr(config)# rmon matrix index 25 port et.1.3
```

rmon nl-matrix-top-n

Purpose

Gathers the top n Network Layer Matrix entries.

Format

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**]

Mode

Configure

Description

The **rmon nl-matrix-top-n** command gathers the top n Network Layer Matrix entries.Before you do this, you should first configure the Application Layer/Network Layer Matrix table using the **rmon hl-matrix** command.

Use the **rmon show nl-matrix-top-n** command to display the top n Network Layer Matrix entries.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the network layer matrix table.

matrix-index <*number*>
Specifies the index into the hl-matrix table. The default is 0.

ratebase terminal-packets | terminal-octets | all-packets | all-octets
Specifies the sorting method:

terminal-packets Sort by terminal packets.

terminal-octets Sort by terminal octets.

all-packets Sort by all packets.

all-octets Sort by all octets.

duration <*number*>
Specifies the duration, in seconds, between reports. The default is 0.

size <*number*>
Specifies the maximum number of matrix entries to include in the report. The default is 150.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To gather the top n Network Layer Matrix entries:

```
ssr(config)# rmon nl-matrix-top-n index 2 matrix-index 25 ratebase all-
packets duration 60 size 100
```

rmon protocol-distribution

Purpose

Configures the RMON 2 Protocol Distribution group.

Format

rmon protocol-distribution index <*index-number*> **port** <*port*> [**owner** <*string*>] [**status enable | disable**]

Mode

Configure

Description

The Protocol Distribution group displays the packets and octets on a protocol and port basis. The **rmon protocol-distribution** command sets various parameters of the RMON 2 Protocol Distribution control table. If default tables were turned on for the Professional group, an entry is created in the Protocol Distribution control table for each available port.

Use the **rmon show protocol-distribution** command to display the protocol distribution.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the Protocol Distribution table.

<*port*>
Specifies the port from which to collect data.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Example

To create an entry in the Protocol Distribution control table:

```
ssr(config)# rmon protocol-distribution index 25 port et.1.3
```

rmon set

Purpose

Configures the Lite, Standard, or Professional RMON groups.

Format

rmon set lite | standard | professional default-tables yes | no

Mode

Configure

Description

You can enable various levels of support (Lite, Standard, or Professional) for RMON groups on a specified set of ports.

Lite adds support for the following RMON 1 groups:

- Ethernet statistics (Etherstats)
- History
- Alarm
- Event

Standard adds support for the following RMON 1 groups:

- Host
- HostTopN
- Matrix
- Filter
- Packet Capture

Professional adds support for the following RMON 2 groups:

- Protocol Directory
- Protocol Distribution

- Address Map
- Network Layer Host
- Network Layer Matrix
- Application Layer Host
- Application Layer Matrix
- User History
- Probe Configuration

A group can consist of a control table and a data table. A control table specifies the statistics to be collected. Each row in the control table specifies the entities for which data is collected, for example, physical ports. The data tables contain the statistics that are collected based on the control table information.

Parameters

lite | standard | professional
Specifies the Lite, Standard, or Professional RMON groups.

default-tables yes
Creates control tables for the following Lite, Standard, or Professional RMON 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 each control table is created for each port on the SSR, with the default owner "monitor".

default-tables no
Removes all control table rows with the owner "monitor". If you wish to save a particular control table row, you must change the owner to a value other than "monitor".

Restrictions

None.

Example

To configure the RMON Lite groups and create default control tables:

```
ssr(config)# rmon set lite default-tables yes
```

rmon set cli-filter

Purpose

Defines filters that can be applied to certain RMON groups during a CLI session.

Format

rmon set cli-filter *<filter-id> <parameter>*

Mode

Configure

Description

You can define filters that CLI users can apply to certain RMON groups. The filters you define are visible to all users that have a Telnet or Console session on the SSR. Each user has the choice of whether or not to apply a particular filter using the **rmon apply cli-filters command**.

RMON CLI filters only affect the output of the following RMON groups:

- Host
- Matrix
- Network Layer Host
- Application Layer Host
- Network Layer Matrix
- Application Layer Matrix
- Protocol Distribution

The **rmon show cli-filters** command displays the RMON CLI filters that have been defined on the SSR.

Parameters

<filter-id>
Is a number between 1 and 65535 that uniquely identifies a CLI filter.

<parameter>

Specifies the parameter on which the filter is set:

src-mac Source MAC Address

dst-mac Destination MAC Address

inpkts In Packets

inoctets In Octets

outpkts out packets

outoctets out Octets

multicast Multicast packets

broadcast Broadcast packets

errors Errors

The following operands can also be used:

and AND

or Or

= Equal to

< Less than

<= Less than or equal to

> Greater than

>= Greater than or equal to

!= Not equal to

(Left bracket

) Right Bracket

src-mac and **dst-mac** can be specified once and the other parameters can be specified multiple times.

Restrictions

None.

Example

To configure an RMON CLI filter on a source MAC address of 123456:123456 and on input packets greater than 1000 and error packets greater than 10 or out packets less than 10000, use the following command:

```
ssr(config)# rmon set cli-filter 3 src-mac 123456:123456 and
((inpkts > 1000 and errors > 10) or (outpkts < 10000))
```

rmon set memory

Purpose

Increases the amount of memory allocated to RMON.

Format

rmon set memory <*number*>

Mode

Enable

Description

RMON allocates memory depending on the number of ports enabled for RMON, the groups that have been configured (Lite, Standard, or Professional) and whether or not default tables have been turned on or off. You can dynamically allocate additional memory to RMON, if needed.

Later, if this additional memory is no longer required, you can reduce the allocation; this change will not take effect until RMON is restarted. This is because memory cannot be freed while RMON is still using it. If the amount of memory specified is less than what RMON has currently allocated, a warning message is displayed and the action is ignored.

Use the **rmon show status** command to display the amount of memory currently allocated to RMON.

Parameters

<*number*>
Specifies the total amount of memory, in Mbytes, to be allocated to RMON. The value can be between 4 and 32.

Note: The number specified is the total number of Mbytes of memory to be allocated; it is not an increment of memory.

Restrictions

None.

Example

To show the amount of memory allocated to RMON:

```
ssr# rmon show status
```

To increase the amount of memory allocated to RMON:

```
ssr# rmon set memory 32
```

rmon set ports

Purpose

Enables RMON on one or more ports.

Format

rmon set ports *<port list>* | **allports**

Mode

Configure

Description

Since RMON uses many system resources, RMON can be enabled on a set of ports. Ports can be dynamically added and removed from the port list. For example, if default tables are turned on for the Lite group and port et.2.1 is then added to the port list, an entry for port et.2.1 is automatically created in the Etherstats and History control tables.

Parameters

<port list>
Specifies the port(s) on which RMON is enabled. Specify **allports** to enable RMON for all ports on the SSR.

Restrictions

None.

Example

To enable RMON on all ports on the SSR:

```
ssr(config)# rmon set ports allports
```

rmon set protocol-directory

Purpose

Specifies the protocol encapsulations that are managed with the Protocol Directory group.

Format

rmon set protocol-directory *<protocol>* **| all-protocols [address-map on | off | na] [host on | off | na] [matrix on | off | na]**

Mode

Configure

Description

The **rmon set protocol-directory** command defines the protocols that are managed with RMON on the SSR.

Parameters

<protocol>
Specifies the protocol encapsulations that are managed with the Protocol Directory group on the SSR. (See Appendix A for a list of protocols supported on the SSR.) Specify **all-protocols** to manage all protocols that are supported on the SSR.

address-map on | off | na
Configures support for the Address Map group for the specified protocol(s).

host on | off | na
Configures support for the Host group for the specified protocol(s).

matrix on | off | na
Configures support for the Matrix group for the specified protocol(s).

Restrictions

The Protocol Directory group is part of the RMON Professional group. To use the **rmon set protocol-directory** command you must enable the RMON Professional group with the **rmon set professional** command.

Example

To configure a protocol encapsulation for the Protocol Directory group:

```
ssr(config)# rmon set protocol-directory all-protocols address-
map on host on matrix on
```

rmon show address-map

Purpose

Displays MAC address to network address bindings for each protocol.

Format

rmon show address-map-logs <*port-list* > | **all-ports**

Mode

Enable

Description

The **rmon show address-map-logs** command displays entries in the RMON 2 Address Map table. Entries in this table are created automatically when default tables are turned on for the Professional group. You can show address bindings for specific ports or for all ports.

Parameters

<*port-list* > | **all-ports**
The port(s) for which you want to display MAC-network address information. Use the keyword **all-ports** to show information for all ports.

Restrictions

This command is only available if you have configured the Professional group and Address Map control table entries exist for the specified port.

Example

To display the address map log table for all ports:

```
ssr# rmon show address-map-logs all-ports
RMON II Address Map Control Table

❶          ❷             ❸               ❹
Port       macAdd        nlAdd           Protocol
----       ------        -----           --------
et.5.1     00001D:CBA3FD 192.100.81.1    ether2.ip-v4
et.5.1     00001D:CBA3FD 192.100.81.1    *ether2.ip-v4
et.5.1     00001D:CBA3FD 10.60.89.88     ether2.ip-v4
et.5.1     00001D:CBA3FD 10.60.89.88     *ether2.ip-v4
et.5.5     00001D:CBA3FD 192.100.81.3    ether2.ip-v4
et.5.5     00001D:CBA3FD 192.100.81.3    *ether2.ip-v4
et.5.5     080020:835CAA 10.60.89.88     ether2.ip-v4
et.5.5     080020:835CAA 10.60.89.88     *ether2.ip-v4
et.5.1     0080C8:C172A6 192.100.81.3    ether2.ip-v4
et.5.1     0080C8:C172A6 192.100.81.3    *ether2.ip-v4
```

Legend:

1. The port on which the MAC address-network address binding was discovered.
2. The MAC address for the binding.
3. The network layer address for the binding.
4. The protocol, as specified in the RMON Protocol Directory for the SSR.

rmon show al-host

Purpose

Shows application layer traffic.

Format

rmon show al-host <*port-list*> | **all-ports** [**summary**]

Mode

Enable

Description

The **rmon show al-host** command shows entries in the RMON 2 Application Layer Host table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Professional group.

If CLI filters have been applied, they will take effect when the Application Layer Host table is displayed. This command shows control rows and their corresponding logs only if there are logs. A control row with no data will not appear in the report.

The Application Layer host group is configured with the **rmon hl-host** command.

Parameters

<*port-list*> | **all-ports**
: The port(s) for which you want to display application layer traffic information. Use the keyword **all-ports** to show traffic information for all the ports.

[**summary**]
: Use the keyword **summary** to display control row summary information only.

Restrictions

This command is only available if you have configured the Professional group and control table entries exist for the specified port.

Example

To show Application Layer Host tables on all ports:

```
ssr# rmon show al-host all-ports
RMON II Application Layer Host Table

Index: 500  Port: et.5.1  Inserts: 9  Deletes: 0  Owner: monitor ❶

❷                ❸       ❹          ❺       ❻         ❼
Address             InPkts   InOctets    OutPkts  OutOctets  Protocol
-------             ------   --------    -------  ---------  --------
10.60.89.88          1080     879418          2        164  *ether2.ip-v4
10.60.89.88          1080     879418          2        164  *ether2.ip-v4.tcp
10.60.89.88          1080     879418          2        164  *ether2.ip-v4.tcp.telnet
192.100.81.1            1        100          1        100  *ether2.ip-v4
192.100.81.1            1        100          1        100  *ether2.ip-v4.icmp
192.100.81.3            3        264       1081     879518  *ether2.ip-v4
192.100.81.3            1        100          1        100  *ether2.ip-v4.icmp
192.100.81.3            2        164       1080     879418  *ether2.ip-v4.tcp
192.100.81.3            2        164       1080     879418  *ether2.ip-v4.tcp.telnet

Index: 504  Port: et.5.5  Inserts: 6  Deletes: 0  Owner: monitor
Address             InPkts   InOctets    OutPkts  OutOctets  Protocol
-------             ------   --------    -------  ---------  --------
10.60.89.88             3        246       1141      92563  *ether2.ip-v4
10.60.89.88             3        246       1141      92563  *ether2.ip-v4.tcp
10.60.89.88             3        246       1141      92563  *ether2.ip-v4.tcp.telnet
192.100.81.3         1141      92563          3        246  *ether2.ip-v4
192.100.81.3         1141      92563          3        246  *ether2.ip-v4.tcp
192.100.81.3         1141      92563          3        246  *ether2.ip-v4.tcp.telnet
```

Legend:

1. The control table entry for this port:

   Index: uniquely identifies the entry in the control table.
   Port: port name.
   Inserts: number of Application Layer Host table entries for this port.
   Deletes: number of Application Layer Host table entries deleted for this port.
   Owner: default owner "monitor."

2. Network address discovered on the port.

3. Number of packets transmitted without errors to the network address for the protocol.

4. Number of octets transmitted without errors to the network address for the protocol.

5. Number of packets transmitted without errors from the network address for the protocol.
6. Number of octets transmitted without errors from the network address for the protocol.
7. The protocol, as specified in the RMON Protocol Directory for the SSR. Note that this shows the destination socket, as well as application/protocol information.

rmon show al-matrix

Purpose

Shows application layer traffic between source and destination addresses.

Format

rmon show al-matrix *<port-list>* **| all-ports [order-by srcdst | dstsrc] [summary]**

Mode

Enable

Description

The **rmon show al-matrix** command shows entries in the RMON 2 Application Layer Matrix table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Professional group.

If CLI filters have been applied, they will take effect when this table is displayed. The control rows and their corresponding logs are displayed only if there are logs. A control row with no data will not appear in the report.

Parameters

<port-list> | **all-ports**
The port(s) for which you want to display application layer traffic information. Use the keyword **all-ports** to show traffic information for all the ports.

srcdst
Orders the logs by source address , then destination address (default).

dstsrc
Orders the logs by destination address, then source address.

summary
Displays control row summary information only.

Restrictions

This command is only available if you have configured the Professional group and control table entries exist for the specified port.

Example

To show the Application Layer Matrix table for all ports:.

```
ssr# rmon show al-matrix all-ports
RMON II Application Layer Host Table

Index: 500  Port: et.5.1  Inserts: 10  Deletes: 0  Owner: monitor ❶

❷               ❸                ❹          ❺      ❻
SrcAddr         DstAddr            Packets     Octets  Protocol
-------         -------            -------     ------  --------
10.60.89.88     192.100.81.3             2        164  *ether2.ip-v4
10.60.89.88     192.100.81.3             2        164  *ether2.ip-v4.tcp
10.60.89.88     192.100.81.3             2        164  *ether2.ip-v4.tcp.telnet
192.100.81.1    192.100.81.3             1        100  *ether2.ip-v4
192.100.81.1    192.100.81.3             1        100  *ether2.ip-v4.icmp
192.100.81.3    10.60.89.88           1181     972211  *ether2.ip-v4
192.100.81.3    10.60.89.88           1181     972211  *ether2.ip-v4.tcp
192.100.81.3    10.60.89.88           1181     972211  *ether2.ip-v4.tcp.telnet
192.100.81.3    192.100.81.1             1        100  *ether2.ip-v4
192.100.81.3    192.100.81.1             1        100  *ether2.ip-v4.icmp

Index: 504  Port: et.5.5  Inserts: 6  Deletes: 0  Owner: monitor
SrcAddr         DstAddr            Packets     Octets  Protocol
-------         -------            -------     ------  --------
10.60.89.88     192.100.81.3          1242     100744  *ether2.ip-v4
10.60.89.88     192.100.81.3          1242     100744  *ether2.ip-v4.tcp
10.60.89.88     192.100.81.3          1242     100744  *ether2.ip-v4.tcp.telnet
192.100.81.3    10.60.89.88              3        246  *ether2.ip-v4
192.100.81.3    10.60.89.88              3        246  *ether2.ip-v4.tcp
192.100.81.3    10.60.89.88              3        246  *ether2.ip-v4.tcp.telnet
```

Legend:

1. The control table entry for this port:

   Index: uniquely identifies the entry in the control table.
   Port: port name.
   Inserts: number of application layer host table entries for this port.
   Deletes: number of application layer host table entries deleted for this port.
   Owner: default owner “monitor.”

2. Source address.

3. Destination address.

4. Number of link layer packets transmitted from the source to the destination without errors for the protocol.
5. Number of octets transmitted from the source to the destination without errors for the protocol.
6. The protocol, as specified in the RMON Protocol Directory for the SSR.

rmon show al-matrix-top-n

Purpose

Reports the top *n* Application Layer Matrix entries, sorted by a specific metric.

Format

rmon show al-matrix-top-n

Mode

Enable

Description

The **rmon show al-matrix-top-n** command shows entries in the RMON 2 Application Layer Matrix Top N table.

Parameters

None.

Restrictions

This command is only available if you have enabled the Professional RMON group and entries exist in the Application Layer Matrix Top N table.

Example

Consider the following command to gather the top n Application Layer Matrix entries:

```
ssr(config)# rmon al-matrix-top-n index 1 matrix-index 500 ratebase all-packets
duration 20 size 5
```

To show the top n entries in the Application Layer Matrix table, as specified by the previous command:

```
ssr# rmon show al-matrix-top-n
RMON II Al Matrix Table

❶    ❷       ❸           ❹      ❺       ❻   ❼              ❽      ❾
Index M-Index RateBase     TimeRem Duration Size StartTime       Reports Owner
    1     500 All-Packets       14       20    5 00D 00H 50M 25S       1 Usama

❿              ⓫           ⓬       ⓭         ⓮        ⓯          ⓰
SrcAddr         DstAddr      PktRate  R-PktRate  OctetRate R-OctetRate Protocol
-------         -------      -------  ---------  --------- ----------- --------
192.100.81.3    10.60.89.88      21         0      19836         0  *ether2.ip-v4.tcp.telnet
192.100.81.3    10.60.89.88       21          0      19836          0  *ether2.ip-v4.tcp
192.100.81.3    10.60.89.88       21          0      19836          0  *ether2.ip-v4
192.100.81.1    192.100.81.3       0          0          0          0  *ether2.ip-v4
192.100.81.3    192.100.81.1       0          0          0          0  *ether2.ip-v4
```

Legend:

1. Index number that identifies this entry in the Application Layer Matrix Top N control table.
2. The Application Layer Matrix table for which the top N report is shown.
3. The parameter on which the entries are sorted.
4. Number of seconds left in the report currently being collected.
5. Number of seconds that this report has collected during the last sampling interval.
6. Maximum number of matrix entries in this report.
7. The time when this report was last started.
8. The number of reports generated by this entry.
9. The entity that configured this entry.
10. Network address of the source host.
11. Network address of the destination host.
12. Number of packets from the source to the destination during the sampling interval.
13. Number of packets from the destination to the source during the sampling interval.
14. Number of octets from the source to the destination during the sampling interval.
15. Number of octets from the destination to the source during the sampling interval.
16. The protocol, as defined in the RMON Protocol Directory group on the SSR.

rmon show alarm

Purpose

Displays configured alarms.

Format

rmon show alarm

Mode

Enable

Description

The **rmon show alarm** command displays the RMON Alarm table.

Parameters

None.

Restrictions

This command is only available if you have configured the Lite group.

Example

To show configured RMON alarms:

```
ssr# rmon show alarm
```

rmon show channels

Purpose

Shows the contents of the Filter Channel table.

Format

rmon show channels

Mode

Enable

Description

The **rmon show channels** command displays the contents of the Filter Channel table.

Parameters

None.

Restrictions

This command is only available if you have configured the Standard group.

Example

To show the contents of the Filter Channel table:

```
ssr# rmon show channels
RMON 1 Channel Table
     No channels defined
```

rmon show cli-filters

Purpose

Displays previously-configured RMON CLI filters.

Format

rmon show cli-filters

Mode

User and Enable.

Description

The **rmon show cli-filters** command displays the RMON CLI filters that have been defined for use on the SSR. Use the **rmon apply cli-filters** command to apply a filter to your current Telnet or Console session.

Parameters

None.

Restrictions

None.

Example

To show RMON CLI filters that are defined on the SSR:

```
ssr> rmon show cli-filters
RMON CLI Filters

❶    ❷
Id   Filter
--   ------
 1   (inpkts >= 0)
 2   (inpkts >= 0 and outoctets >= 0)
 3   srcmac 222222222222 and (outoctets >= 0)
You have selected a filter: (inpkts >= 0) ❸
```

Legend:

1. The filter ID. You use this value to apply a filter with the **rmon apply cli-filters** command.
2. The filter parameters that were specified with the **rmon set cli-filter** command.
3. This shows the parameters of the filter that is currently applied to your Telnet or Console session.

rmon show etherstats

Purpose

Displays Ethernet statistics for one or more ports.

Format

rmon show etherstats <*port-list*> | **all-ports**

Mode

Enable

Description

The **rmon show etherstats** command displays entries in the Ethernet table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Lite group.

Parameters

<*port-list*> | **all-ports**
The port(s) for which you want Ethernet statistics displayed. Use the keyword **all-ports** to show Ethernet statistics on all ports.

Restrictions

This command is only available if you have configured the Lite group.

Example

To display Ethernet statistics on a specified port:

```
ssr# rmon show etherstats et.5.1
RMON I Ethernet Statistics Table
Index: 502  Port: et.5.1  Owner: monitor (1)
-----
RMON EtherStats                 Total
---------------                 -----
Octets                       83616016 (2)
Unicast Frames                  86185 (3)
Broadcast Frames                    0 (4)
Multicast Frames                    0 (5)
Collisions                          0 (6)
64 Byte Frames                    292 (7)
65-127 Byte Frames              16625
128-255 Byte Frames              6145
256-511 Byte Frames              4520
512-1023 Byte Frames             7992
1024-1518 Byte Frames           50611
```

Legend:

1. The EtherStats control table entry for this port:

   Index: uniquely identifies this entry.
   Port: port et.5.1.
   Owner: default owner "monitor."

2. Number of octets of data received on the network.
3. Number of good frames received that were directed to a Unicast address.
4. Number of good frames received that were directed to a broadcast address.
5. Number of good frames received that were directed to a multicast address.
6. Number of collisions on this Ethernet segment.
7. Number of good and bad frames received, for various frame size ranges.

rmon show events

Purpose

Displays configured events and logs of triggered events.

Format

rmon show events

Mode

Enable

Description

The **rmon show events** command displays configured events and the logs, if any, of triggered events.

Parameters

None.

Restrictions

This command is only available if you have configured the Lite group.

Example

To show RMON events and logs:

```
ssr# rmon show events
RMON I Event table

 ❶    ❷    ❸              ❹                        ❺
Index Type Community       Description               Owner
    1 log  public          Log Only                  Usama
No event logs found ❻
Index Type Community       Description               Owner
    2 both private         Log & Trap                Usama
No event logs found
```

Legend:

1. Index number that identifies this entry in the Event table.
2. Type of event: log, trap, or both log and trap.
3. Community string used for this event.
4. User-defined description of this event.
5. Owner of this event entry.

rmon show filters

Purpose

Shows the contents of the Filters table.

Format

rmon show filters

Mode

Enable

Description

The rmon show filters command show the contents of the Filter table.

Parameters

None.

Restrictions

This command is only available if you have configured the Standard group.

Example

To show the contents of the Filter table:

```
ssr# rmon show filters
RMON 1 Filter Table
     No filters defined
```

rmon show history

Purpose

Shows statistics over a period of time.

Format

rmon show history <*port-list*> | **all-ports**

Mode

Enable

Description

The **rmon show history** command displays statistical samples that are stored in the RMON History group. Entries in this table are created automatically when default tables are turned on for the Lite group.

Parameters

<*port-list*> | **all-ports**
The port(s) for which the history is to be displayed. Use the keyword **all-ports** to show history information on all the ports.

Restrictions

This command is only available if you have configured the Lite group.

Example

To display history information for a specific port:

```
ssr# rmon show history et.5.1
RMON I History Table

❶    ❷          ❸              ❹      ❺
Index Port       Interval(secs) Buckets Owner
  502 et.5.1                   300  50/50  monitor

  ❻    ❼                    ❽      ❾        ❿     ⓫     ⓬     ⓭
  Index SysUpTime            Octets Packets   Bcst   Mcst  Colls  %Util  Other
    213 00D 17H 45M 47S      318114     336      0      0      0      0      0
    214 00D 17H 50M 47S      323928     341      0      0      0      0      0
    215 00D 17H 55M 48S      323586     335      0      0      0      0      0
    216 00D 18H 00M 49S      317186     320      0      0      0      0      0
    217 00D 18H 05M 49S      323470     333      0      0      0      0      0
                                          .
                                          .
                                          .
    258 00D 21H 31M 03S      322264     312      0      0      0      0      0
    259 00D 21H 36M 03S      327944     315      0      0      0      0      0
    260 00D 21H 41M 04S      333138     309      0      0      0      0      0
    261 00D 21H 46M 06S      327782     312      0      0      0      0      0
    262 00D 21H 51M 07S      332268     294      0      0      0      0      0
```

Legend:

1. Index number that identifies the entry for this port in the History control table.
2. Port name.
3. Interval (in seconds) for data samples for each data bucket.
4. The actual number of buckets/the requested number of buckets.
5. Owner of this entry “monitor” (default).
6. Index number for this data bucket.
7. Time at which the sample was measured.
8. Total number of octets received on the network.
9. Number of packets received during the sampling period.
10. Number of good packets received during the sampling interval that were directed to a broadcast address.
11. Number of good packets received during the sampling interval that were directed to a multicast.
12. The number of collisions on this Ethernet segment during the sampling interval (best estimate).
13. The percentage of the network being utilized (best estimate).

rmon show host-top-n

Purpose

Displays the top *n* hosts.

Format

rmon show host-top-n

Mode

Enable

Description

The **rmon show host-top-n** command displays a report of the top hosts for a specified statistic. Note that the Host Top N report runs once. To run the reports again via the CLI, the control row must be disabled and then enabled. If the report has already been run, the Time Remaining field is set to zero. Otherwise, the Time Remaining field will be decremented until the report is run.

Restrictions

This command is only available if you have configured the Standard group and Host Top N control table entries exist.

Example

Consider the following command to gather the top n Host entries:

```
ssr(config)# rmon host-top-n index 1 host-index 500 base out-octets duration
20 size 5
```

To display the Host Top N report, as specified by the previous command:

```
ssr# rmon show host-top-n
RMON I HostTopN Table

❶    ❷         ❸             ❹       ❺      ❻       ❼               ❽
Index HostIndex RateBase      TimeRem Duration Buckets StartTime       Owner
1       500     Out-Octets    0            20    5/5   00D 00H 39M 29S  Usama

❾                          ❿
Address                    Rate
-------                    ----
0080C8:C172A6              19911
00001D:CBA3FD                  0
```

Legend:

1. Index number that identifies this entry in the Host Top N control table.
2. Index number that identifies the Host control table entry.
3. The parameter used to order the list of top “n” entries.
4. Number of seconds left in the report currently being collected.
5. Number of seconds that this report has collected during the last (or current) sampling interval.
6. Maximum number of hosts requested for the Top N table/maximum number of hosts in the Top N table.
7. The time of the sampling.
8. The owner of this entry.
9. The host address.
10. The value of the statistic for the host address.

rmon show hosts

Purpose

Shows statistics about the hosts discovered on the network.

Format

rmon show hosts <*port-list*> | **all-ports** [**summary**]

Mode

Enable

Description

The **rmon show hosts** command displays entries in the Hosts table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Standard group.

If CLI filters have been applied, they will take effect when the Host table is displayed. This command will display control rows and their corresponding logs only if there are logs. A control row that has no data is not displayed.

Parameters

<*port-list*> | **all-ports**
The port(s) for which host information is to be shown. Use the keyword **all-ports** to show host information on all the ports.

summary
Use the keyword **summary** to show a summary of all control table rows with the number of logs in each row.

Restrictions

This command is only available if you have configured the Standard group and control table entries exist for the specified port.

Example

To show host information for a specific port:

```
ssr# rmon show hosts et.5.1
RMON I Host Table
Index: 502  Port: et.5.1  Owner: monitor ❶

❷                ❸       ❹          ❺       ❻            ❼     ❽
Address          InPkts   InOctets    OutPkts  OutOctets      Bcst   Mcst
-------          ------   --------    -------  ---------      ----   ----
00001D:CBA3FD     88917   88436760      62132   5 095029      0         0
0080C8:C172A6     62132    5095029      88920   88437062      0         0
```

Legend:

1. Host control table information for this port:

   Index: number that identifies the entry for this port in the table.
   Port: port name.
   Owner: the default owner "monitor."

2. MAC address of the discovered host.
3. Number of good packets transmitted to this address.
4. Number of good octets transmitted to this address.
5. Number of good packets transmitted from this address.
6. Number of good octets transmitted from this address.
7. Number of good packets transmitted by this address that were directed to a broadcast address.
8. Number of good packets transmitted by this address that were directed to a multicast address.

To show a summary of host information:

```
ssr# rmon show all-ports summary
RMON I Host Table Summary

❶    ❷        ❸      ❹     ❺      ❻
Index Data Rows Port    Status Mode    Owner
----- --------- ----    ------ ----    -----
  500         1 et.5.1  Up     Address monitor
  501         1 et.5.2  Up     Address monitor
  502         0 et.5.3  Down   Flow    monitor
  503        17 et.5.4  Up     Flow    monitor
  504         0 et.5.5  Down   Flow    monitor
  505         0 et.5.6  Down   Flow    monitor
  506         0 et.5.7  Down   Flow    monitor
  507         0 et.5.8  Down   Flow    monitor
```

Legend:

1. Index number that identifies this entry in the Host control table.
2. Number of data rows associated with this index number.
3. Port.
4. Current state of the port.
5. Source of the data for this entry.
6. Owner of this entry.

rmon show matrix

Purpose

Shows statistics for source-destination address pairs.

Format

rmon show matrix <*port-list*> | **all-ports [summary] [order-by srcdst | dstsrc]**

Mode

Enable

Description

The **rmon show matrix** command displays entries in the Matrix table. Entries in this table are automatically created when default tables are turned on for the Standard group.

If CLI filters have been applied, they will take effect when the Matrix table is displayed. This command will display control r ows and their corresponding logs only if there are logs. A control row that has no data is not displayed.

Parameters

<*port-list*> | **all-ports**
The port(s) for which you want to display information. Use the keyword **all-ports** to show matrix information on all the ports.

summary | order by
Use the keyword **summary** to display the control rows only. Use the keyword **order-by** to display entries by source/destination or by destination/source.

srcdst | dstsrc
Use the keyword **srcdst** to display the entries by source/destination. Use the keyword **dstsrc** to display entries by destination/source.

Restrictions

This command is only available if you have configured the Standard group.

Example

To show statistics for source-destination address pairs:

```
ssr# rmon show matrix all-ports
RMON I Matrix Table

Port: et.5.1  Index: 500  Owner: monitor ❶

❷                    ❸                    ❹          ❺
SrcAddr                DstAddr                Packets     Octets
-------                -------                -------     ------
00001D:CBA3FD          0080C8:C172A6            3           264
0080C8:C172A6          00001D:CBA3FD            4           346

Port: et.5.5  Index: 504  Owner: monitor
SrcAddr                DstAddr               Packets      Octets
-------                -------               -------      ------
00001D:CBA3FD          080020:835CAA           3           246
080020:835CAA          00001D:CBA3FD           2           164
```

Legend:

1. The Matrix control table entry for this port:

   Port: the name of the port.
   Index: the index number for this port in the Matrix table.
   Owner: default "monitor."

2. Source MAC address.

3. Destination MAC address.

4. Number of packets transmitted from the source to the destination address, including bad packets.

5. Number of octets transmitted from the source to the destination address.

To show control row summary statistics:

```
ssr# rmon show matrix all-ports summary
RMON I Matrix Table Summary
 Index  Data Rows  Port   Status   Mode   Owner
-----   ---------  ----   ------   ----   ----
500        0      et.1.1   Up     Address monitor
501        0      et.1.2   Down   Address monitor
502        0      et.1.3   Down   Address monitor
503        0      et.1.4   Up     Address monitor
504        0      et.1.5   Down   Address monitor
505        0      et.1.6   Down   Address monitor
506        0      et.1.7   Down   Address monitor
507        0      et.1.8   Up     Address monitor
508        0      gi.4.1   Up     Address monitor
509        0      gi.4.2   Up     Address monitor
510        0      et.7.1   Up     Address monitor
511        0      et.7.2   Down   Address monitor
512        0      et.7.3   Down   Address monitor
513        0      et.7.4   Down   Address monitor
514        0      et.7.5   Down   Address monitor
515        0      et.7.6   Down   Address monitor
516        0      et.7.7   Down   Address monitor
517        0      et.7.8   Down   Address monitor
25         0      et.1.3   Down   Address
ssr#
```

rmon show nl-host

Purpose

Shows the amount of traffic to and from each network address.

Format

rmon show nl-host <*port-list*> | **all-ports** [**summary**]

Mode

Enable

Description

The **rmon show nl-host** command shows entries in the RMON 2 Network Layer Host table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Professional group.

If CLI filters have been applied, they will take effect when the Network Layer host table is displayed. This command shows control rows and their corresponding logs only if there are logs. A control row with no data will not appear in the report.

Parameters

<*port-list*> | **all-ports**
The port(s) for which you want to display traffic information. Use the keyword **all-ports** to show information on all the ports.

summary
Use the keyword **summary** to display control row summary information only.

Restrictions

This command is only available if you have configured the Professional RMON group and control table entries exist for the specified port.

Example

To display the network layer host table for all ports:

```
ssr# rmon show nl-host all-ports

RMON II Network Layer Host Table

Index: 500  Port: et.5.1  Inserts: 3  Deletes: 0  Owner: monitor ❶

❷               ❸       ❹         ❺       ❻         ❼
Address           InPkts   InOctets    OutPkts  OutOctets  Protocol
-------           ------   --------    -------  ---------  --------
10.60.89.88         1159     952300          2        164  *ether2.ip-v4
192.100.81.1           1        100          1        100  *ether2.ip-v4
192.100.81.3           3        264       1160     952400  *ether2.ip-v4

Index: 504  Port: et.5.5  Inserts: 2  Deletes: 0  Owner: monitor
Address           InPkts   InOctets    OutPkts  OutOctets  Protocol
-------           ------   --------    -------  ---------  --------
10.60.89.88            3        246       1220      98962  *ether2.ip-v4
192.100.81.3        1220      98962          3        246  *ether2.ip-v4
```

Legend:

1. The control table entry for this port:

   Index: index number that identifies this entry in the hl host control table.
   Port: name of port.
   Inserts: number of inserts in the network layer host table for this entry.
   Deletes: number of deletions in the network layer host table for this entry.
   Owner: the entity that configured this entry.

2. The network address.
3. Number of packets received by this network address.
4. Number of octets received by this network address.
5. Number of packets sent by this network address.
6. Number of octets sent by this network address.
7. The protocol, as defined in the RMON Protocol Directory for the SSR. Note that this shows the network layer protocol encapsulations only. If you want to see application/protocol information, such as the destination socket, use the **rmon show al-host** command.

rmon show nl-matrix

Purpose

Shows information about the traffic between network address pairs.

Format

rmon show nl-matrix <*port-list*> | **all-ports [order-by srcdst | dstsrc] [summary]**

Mode

Enable

Description

The **rmon show nl-matrix** command shows entries in the Network Layer Matrix table for one or more ports. Entries in this table are created automatically when default tables are turned on for the Professional group**.**

If CLI filters have been applied, they will take effect when this table is displayed. The control rows and their corresponding logs are displayed only if there are logs. A control row with no data will not appear in the report.

Parameters

<*port-list*> | **all-ports**
The port(s) for which you want to display network layer traffic information. Use the keyword **all-ports** to show information for all ports.

order-by srcdst
Orders the logs by source address, then destination address (default).

order-by dstsrc
Orders the logs by destination address, then source address.

summary
Use the keyword **summary** to display control row summary information only.

Restrictions

This command is only available if you have configured the Professional group and control table entries exist for the specified port.

Example

To show the Network Layer Matrix table for all ports:

```
ssr# rmon show nl-matrix all-ports
RMON II Network Layer Matrix Table

Index: 500  Port: et.5.1  Inserts: 4  Deletes: 0  Owner: monitor ❶

❷              ❸                ❹          ❺      ❻
SrcAddr         DstAddr            Packets     Octets  Protocol
-------         -------            -------     ------  --------
10.60.89.88     192.100.81.3             2        164  *ether2.ip-v4
192.100.81.1    192.100.81.3             1        100  *ether2.ip-v4
192.100.81.3    10.60.89.88           1241    1025436  *ether2.ip-v4
192.100.81.3    192.100.81.1             1        100  *ether2.ip-v4

Index: 504  Port: et.5.5  Inserts: 2  Deletes: 0  Owner: monitor
SrcAddr         DstAddr            Packets     Octets  Protocol
-------         -------            -------     ------  --------
10.60.89.88     192.100.81.3          1302     105604  *ether2.ip-v4
192.100.81.3    10.60.89.88              3        246  *ether2.ip-v4
```

Legend:

1. The control table entry for this port:

   Index: index number that identifies this entry in the control table.
   Port: name of port.
   Inserts: number of inserts in the Network Layer Matrix table for this entry.
   Deletes: number of deletions in the Network Layer Matrix table for this entry.
   Owner: the entity that configured this entry.

2. Source network address.
3. Destination network address.
4. Number of packets transmitted without error from the source to the destination.
5. Number of octets transmitted without error from the source to the destination.
6. The protocol, as specified in the RMON Protocol Directory for the SSR.

rmon show nl-matrix-top-n

Purpose

Reports the top *n* Network Layer Matrix entries, sorted by a specific metric.

Format

rmon show nl-matrix-top-n

Mode

Enable

Description

The rmon show nl-matrix-top-n command shows entries in the RMON 2 Network Layer Matrix Top N table.

Parameters

None.

Restrictions

This command is only available if you have configured the Professional group and entries exist in the Network Layer Matrix Top N table.

Example

Consider the following command to gather the top n Network Layer Matrix entries:

```
ssr(config)# rmon nl-matrix-top-n index 1 matrix-index 500 ratebase all-octets duration 20
size 5
```

To show the top n entries in the Network Layer Matrix table, as specified by the previous command:

```
ssr# rmon show nl-matrix-top-n
RMON II Nl Matrix Table

❶    ❷      ❸           ❹      ❺       ❻   ❼              ❽      ❾
Index M-Index RateBase     TimeRem Duration Size StartTime       Reports Owner
    1     500 Octets            20       20    5 00D 00H 51M 37S       1 Usama

❿              ⓫                ⓬       ⓭         ⓮        ⓯          ⓰
SrcAddr         DstAddr            PktRate  R-PktRate  OctetRate R-OctetRate Protocol
-------         -------            -------  ---------  --------- ----------- --------
192.100.81.3    10.60.89.88             23          0      19986           0 *ether2.ip-v4
192.100.81.1    192.100.81.3             0          0          0           0 *ether2.ip-v4
192.100.81.3    192.100.81.1             0          0          0           0 *ether2.ip-v4
10.60.89.88     192.100.81.3             0         23          0       19986 *ether2.ip-v4
```

Legend:

1. Index number that identifies this entry in the network layer Matrix Top N control table.
2. The Network Layer Matrix table for which the top N report is shown.
3. The parameter on which the entries are sorted.
4. Number of seconds left in the report currently being collected.
5. Number of seconds that this report has collected during the last sampling interval.
6. Maximum number of matrix entries in this report.
7. The time when this report was last started.
8. The number of reports generated by this entry.
9. The entity that configured this entry.
10. Network address of the source host.
11. Network address of the destination host.
12. Number of packets from the source to the destination during the sampling interval.
13. Number of packets from the destination to the source during the sampling interval.
14. Number of octets from the source to the destination during the sampling interval.
15. Number of octets from the destination to the source during the sampling interval.
16. The protocol, as defined in the RMON Protocol Directory for the SSR.

rmon show packet-capture

Purpose

Shows packets captured after flowing through a channel.

Format

rmon show packet-capture

Mode

Enable

Description

The **rmon show packet-capture** command shows the buffer table for captured packets. Before you use this command, first configure the Filter Channel group using the **rmon channel index** command. Then use the **rmon capture** command to configure the Packet Capture group which allows packets to be captured after they have flowed through a channel.

Parameters

None.

Restrictions

This command is only available if you have enabled the Standard RMON groups.

rmon show probe-config

Purpose

Shows the configuration of the SSR for interaction with other RMON devices.

Format

probe-config [basic] [net-config] [trap-dest]

Mode

Enable

Description

The **rmon show probe-config** command shows entries in the RMON 2 Probe Configuration table.

Parameters

basic	Shows basic probe configuration information.
net-config	Shows network configuration table.
trap-dest	Shows trap destination table.

Restrictions

This command is only available if you have configured the Professional group.

rmon show protocol-directory

Purpose

Displays the protocols that the SSR can monitor with RMON.

Format

rmon show protocol-directory <*protocol*> | **all-protocols**

Mode

Enable

Description

The **rmon show protocol-directory** command displays the protocol encapsulations that are defined in the RMON 2 Protocol Directory group for the SSR.

Parameters

<*protocol*> | **all-protocols**
The specific protocol encapsulation that is managed with the RMON 2 Protocol Directory group. (See Appendix A for protocol encapsulations that are supported on the SSR.) Use the keyword **all-protocols** to display all protocol encapsulations that are managed with the Protocol Directory group.

Restrictions

This command is only available if you have configured the Professional group.

Example

To show all protocol encapsulations that are managed with the Protocol Directory group:

:

```
ssr# rmon show protocol-directory all-protocols
RMON II Protocol Directory Table

Last Change: 00D 00H 00M 00S
Index AddrMap Host Matrix Status Protocol
1      Off    Off  Off    Active ether2
2      NA     Off  Off    Active idp
3      NA     Off  Off    Active ip-v4
4      NA     Off  Off    Active chaosnet
5      NA     Off  Off    Active arp
6      NA     Off  Off    Active rarp
7      NA     Off  Off    Active vip
8      NA     Off  Off    Active vloop
9      NA     Off  Off    Active vloop2
10     NA     Off  Off    Active vecho
11     NA     Off  Off    Active vecho2
12     NA     Off  Off    Active ipx
13     NA     Off  Off    Active netbios-3com
14     NA     Off  Off    Active atalk
15     NA     Off  Off    Active aarp

...
```

NOTE: The example above shows a partial listing only.

rmon show protocol-distribution

Purpose

Shows the octets and packets detected for different protocols on a network segment.

Format

rmon show protocol-distribution <*port-list*> | **all-ports**

Mode

Enable

Description

The **rmon show protocol-distribution** command displays the RMON 2 Protocol Distribution table. This table contains a list of protocols, defined in the RMON 2 Protocol Directory, that are discovered by the SSR. Entries in this table are created automatically when default tables are turned on for the Professional group. If you delete an entry in the Protocol Directory, then entries in this table associated with the deleted protocol are also deleted.

If CLI filters have been applied, they will take effect when the Protocol Distribution table is displayed.

Parameters

<*port-list*> | **all-ports**
The port(s) for which you want to show protocol distribution. Use the keyword **all-ports** to show protocol distribution information on all the ports.

Restrictions

This command is only available if you have configured the Professional group.

Example

To show the RMON 2 Protocol Distribution table:

:

```
ssr(config)# rmon show protocol-distribution all-ports
RMON II Protocol Distribution Table

Index: 508  Port: gi.4.1  Owner: monitor
Pkts Octets Protocol
---- ------ --------
3312 304550 ether2
3312 304550 ip-v4
2459 234564 icmp
 853  69986 tcp
 853  69986 telnet
```

rmon show status

Purpose

Displays RMON status, groups, enabled ports, and memory utilization.

Format

rmon show status

Mode

Enable

Description

The **rmon show status** command shows whether RMON is enabled, the RMON groups that are configured, the ports on which RMON is enabled, and the memory allocated and used by RMON.

Parameters

None.

Example

To show RMON status:

```
ssr# rmon show status
RMON Status
-----------
* RMON is ENABLED ❶
* RMON initialization successful.

+--------------------------+
| RMON Group Status        | ❷
+-------+--------+---------+
| 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 ❸

RMON Memory Utilization ❹
-----------------------
        Total Bytes Available:   48530436

Total Bytes Allocated to RMON:    4000000
             Total Bytes Used:    2637872
             Total Bytes Free:    1362128
```

Legend:

1. When the SSR is booted, RMON is off by default. RMON is enabled with the **rmon enable** command.
2. Shows which RMON group (Lite, Standard, or Professional) is configured and whether default control tables are turned on.
3. Shows the ports on which RMON is enabled.
4. Shows RMON memory utilization. You can adjust the amount of memory allocated to RMON with the **rmon set memory** command.

rmon show user-history

Purpose

Shows user-defined collection of historical information from MIB objects on the SSR.

Format

rmon show user-history

Mode

Enable

Description

The **rmon show user-history** command shows the User History table.

Parameters

None.

Restrictions

This command is only available if you have configured the Professional group.

rmon user-history-apply

Purpose

Applies a specified group to the User History control table.

Format

rmon user-history-apply <*groupname*> **to** <*user-history-index*>

Mode

Configure

Description

The **rmon user-history-apply** command applies all objects in the group created with the **rmon user-history-objects** command to the row in the User History control table. If the number of objects specified in the control row is greater than those in the group, the remaining OIDs are set to 0.0. If the number of objects specified in the control row is less than those in the group, the remaining are discarded.

Parameters

<*groupname*>
Is the name of a group of objects that has been created with the **rmon-user-history-objects** command.

<*user-history-index*>
Specifies the row in the User History control table.

Restrictions

None.

rmon user-history-control

Purpose

Monitors a group of objects (OIDs) over a period of time.

Format

rmon user-history-control index <*index-number*> **objects** <*number*> **samples** <*number*> **interval** <*number*> [**owner** <*string*>] [**status enable** | **disable**]

Mode

Configure

Description

The **rmon user-history-control** command monitors the group of objects that are defined with the **rmon user-history-objects** command. This command creates an entry in the User History control table.

Use the **rmon show user-history** command to display the User History table.

Parameters

<*index-number*>
Is a number between 1 and 65535 that uniquely identifies a row in the user history control table.

objects <*number*>
Specifies the number of MIB objects to be collected.

samples <*number*>
Specifies the number of discrete time intervals over which data is to be saved.

interval <*number*>
Specifies the interval, in seconds, between samples.

owner <*string*>
Specifies the owner of the event; for example, an IP address, machine name or person's name.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

rmon user-history-objects

Purpose

Defines a group of objects (OIDs).

Format

rmon user-history-objects <*groupname*> **variable** <*oid*> **type absolute | delta** [**status enable | disable**]

Mode

Configure

Description

The **rmon user-history-objects** command defines the group of objects that can be monitored with the **rmon user-history-control** command. This command creates a group with a single OID as a member of the group. To add several objects to the group, you need to issue multiple **user-history-objects** commands. Each object appears as a separate row in the User History control table.

Parameters

<*groupname*>
Is the name of the group of objects.

variable <*oid*>
Specifies the object identifier to be monitored.

type absolute | delta
Specifies the method of sampling for the selected variable.

interval <*number*>
Specifies the interval, in seconds, between samples.

status enable | disable
Enables or disables this matrix. The default is enable.

Restrictions

None.

Chapter 49

save Command

The **save** command saves the configuration changes you have entered during the current CLI session. You can save the configuration commands in the scratchpad to the active configuration, thus activating changes. You then can save the active changes to the Startup configuration.

Format

save active | startup

Mode

Configure

Note: If you are in Enable mode, you still can save the active configuration changes to the Startup configuration file by entering the **copy active to startup** command.

Description

Saves configuration changes.

- If you use the **active** keyword, uncommitted changes in the scratchpad are activated. The SSR accumulates configuration commands in the scratchpad until you activate them or clear them (or reboot). When you activate the changes, the SSR runs the commands.
- If you use the **startup** keyword, the configuration of the running system is saved in the Startup configuration file and re-instated by the server the next time you reboot.

Parameters

active | startup Specifies the destination for the configuration commands you are saving.

Restrictions

None.

Chapter 50

sfs Commands

The sfs commands set and display the following parameters:

- Cabletron Discovery Protocol (CDP) parameters

Command Summary

Table 36 lists the port commands. The sections following the table describe the command syntax.

Table 36. sfs commands

sfs enable cdp-hello <*port-list*> | **all-ports**
sfs set cdp-hello transmit-frequency
sfs show cdp-hello port-status <*port-list*> | **all-ports**
sfs show cdp-hello transmit-frequency

sfs enable cdp-hello

Purpose

Enabled the sending of CDP Hello packets.

Format

sfs enable cdp-hello <*port-list*> | **all-ports**

Mode

Configure

Description

The **sfs enable cdp-hello** command enables the sending of CDP (Cabletron Discovery Protocol) Hello packets. These are special packets sent out periodically by the router to announce itself to other Cabletron devices or applications. CDP Hello packets can be enabled to be sent out to all available ports or selected ports only.

Parameters

<*port-list*> | **all-ports**	Specifies the ports you want to enable CDP Hello packets. The **all-ports** keyword enables CDP Hello packets for all the SSR ports.

Restrictions

None.

Examples

To enable the sending of CDP Hello packets on port 3 of slot 1:

```
ssr(config)# sfs enable cdp-hello et.1.3
```

To send CDP Hello packets on all ports:

```
ssr(config)# sfs enable cdp-hello all-ports
```

sfs set cdp-hello transmit-frequency

Purpose

Specify how often CDP Hello packets should be sent.

Format

sfs set cdp-hello transmit-frequency <*secs*>

Mode

Configure

Description

The **sfs set cdp-hello transmit-frequency** command specifies how often CDP Hello packets should be sent. The interval is specified in seconds. The default transmit frequency is one packet every 5 seconds.

Parameters

<*secs*>	Specifies the interval in seconds between the transmission of CDP Hello packets. Acceptable value is 1-300. Default is 5 seconds.

Restrictions

None.

Examples

To set the transmit frequency to 10 seconds:

```
ssr(config)# sfs set cdp-hello transmit-frequency 10
```

sfs show cdp-hello port-status

Purpose

Display CDP Hello status of a port.

Format

sfs show cdp-hello port-status <*port-list*> | **all-ports**

Mode

Enable

Description

The **sfs show cdp-hello port-status** command displays CDP Hello information of SSR ports.

Parameters

<*port-list*>	**all-ports** Specifies the ports for which you want to display information. The **all-ports** keyword displays the selected information for all the SSR ports.

Restrictions

None.

Examples

To display CDP Hello status on all SSR ports:

```
ssr# sfs show cdp-hello port-status all-ports
```

sfs show cdp-hello transmit-frequency

Purpose

Display the transmit frequency of CDP Hello packets.

Format

sfs show cdp-hello transmit-frequency

Mode

Enable

Description

The **sfs show cdp-hello transmit-frequency** command display the transmit frequency of CDP Hello packets on the SSR.

Parameters

None.

Restrictions

None.

Examples

To display the transmit frequency of CDP Hello packets:

```
ssr# sfs show cdp-hello transmit-frequency
```

Chapter 51

show Command

Purpose

The **show** command displays the configuration of your running system.

Format

show

Mode

Configure

Description

The **show** command displays the configuration of your running system as well as any non-committed changes in the scratchpad. Each CLI command is preceded with a number. This number can be used with the **negate** command to negate one or more commands. If you see the character **E** (for Error) immediately following the command number, it means the command did not execute successfully due of an earlier error condition. To get rid of the command in error, you can either negate it or fix the original error condition.

When viewing the active configuration file, the CLI displays the configuration file command lines with the following possible annotations:

- Commands without errors are displayed without any annotation.
- Commands with errors are annotated with an “E”.

- If a particular command has been applied such that it can be expanded on additional interfaces/modules, then it is annotated with a "P". For example, if you enable STP on all ports in the current system, but the SSR contains only one module, then that particular command will be extended to all modules when they have been added to the SSR.

  A command like **stp enable et.*.*** would be displayed as follows:

  ```
  P: stp enable et.*.*
  ```

  indicating that it is only partially applied. If you add more modules to the SSR at a later date and then update the configuration file to encompass all of the available modules in the SSR, then the "P:" portion of the above command line would disappear when displaying this configuration file.

  If a potentially partial command, which was originally configured to encompass all of the available modules on the SSR, becomes only partially activated (after a hotswap or some such chassis reconfiguration), then the status of that command line will automatically change to indicate a partial completion status, complete with "P:".

Note: Commands with no annotation or annotated with a "P:" are not in error.

Parameters

None.

Restrictions

None.

Examples

The following command shows when the running system was last modified (Jan 15) and from where (Console). It also shows that there are seven commands currently used to configure the system. In addition, command #7 is shown as having an error condition (**E**) possibly because the VLAN name *abc* is not defined. The actual cause of the error should

have been displayed earlier when the command was first committed to the running system. This is the time when the error was first detected.

```
ssr(config)# show
!
! Last modified from Console on Fri Jan 15 10:33:30 1999
!
1 : vlan create IP1 ip
2 : vlan create IP2 ip
3 : vlan create IP3 ip
!
4 : interface create ip ssr0 address-netmask 10.1.13.1/24 vlan IP1
5 : interface create ip ssr1 address-netmask 10.1.11.1/24 vlan IP2
6 : interface create ip ssr2 address-netmask 10.1.12.1/24 vlan IP3
7E: interface create ip ssr3 address-netmask 10.1.63.12/24 vlan abc
```

To correct the error condition for command #7, a new command is entered to create a VLAN called IP4. The **show** command now displays not only the active configuration but also non-committed commands in the scratchpad.

```
ssr(config)# show
!
! Last modified from Console on Fri Jan 15 10:33:30 1999
!
1 : vlan create IP1 ip
2 : vlan create IP2 ip
3 : vlan create IP3 ip
!
4 : interface create ip ssr0 address-netmask 10.1.13.1/24 vlan IP1
5 : interface create ip ssr1 address-netmask 10.1.11.1/24 vlan IP2
6 : interface create ip ssr2 address-netmask 10.1.12.1/24 vlan IP3
7E: interface create ip ssr3 address-netmask 10.1.63.12/24 vlan IP4

******** Non-committed changes in Scratchpad ********
1*: vlan create IP4 ip
```

The following series of command line examples shows the use of the “partial” flag/annotation when viewing configuration file command line(s).

Suppose you have created VLAN “x” and added ports et.1.1 and et.2.1 to that VLAN. The display in the configuration file would look like this:

```
vlan add ports et1.1 et2.1 to x
```

Now, you decide to hotswap module 2 out of the system. The command line display then looks like the following:

```
P: vlan add ports et.1.1 et.2.1 to x
```

Suppose you now hotswap module 1 out of the system meaning that neither of the ports you configured for this command line exist in the SSR. You will see an “error” indicator/annotation in the command line display as follows:

```
E: vlan add ports et.1.1 et.2.1 to x
```

Certain commands are always shown with a “partial” annotation in their configuration file command lines, as they are always able to be expanded. The following command line gives an example of this:

```
P: ip disable proxy-arp interface all
```

Since this particular command applies to all interfaces, it encompasses all existing interfaces as well as any that might be configured in the future.

Chapter 52

smarttrunk Commands

The **smarttrunk** commands let you display and set parameters for SmartTRUNK ports. SmartTRUNK ports are groups of ports that have been logically combined to increase throughput and provide link redundancy.

Command Summary

Table 37 lists the **smarttrunk** commands. The sections following the table describe the command syntax.

Table 37. smarttrunk commands

smarttrunk add ports <*port list*> **to** <*smarttrunk*>
smarttrunk clear load-distribution <*smarttrunk*>
smarttrunk create <*smarttrunk*> **protocol** <*protocol*>
smarttrunk set load-policy on <*smarttrunk*> <*load-policy*>
smarttrunk show <*option*>

smarttrunk add ports

Purpose

Adds physical ports to a SmartTRUNK.

Format

smarttrunk add ports <*port list*> **to** <*smarttrunk*>

Mode

Configure

Description

The **smarttrunk add ports** command allows you to add the ports specified in <*port list*> to a SmartTRUNK. The SmartTRUNK must already have been created with the **smarttrunk create** command. The ports in the SmartTRUNK must be set to full duplex.

Parameters

<*port list*>	Is one or more ports to be added to an existing SmartTRUNK. All the ports in the SmartTRUNK must be connected to the same destination.
<*smarttrunk*>	Is the name of an existing SmartTRUNK.

Restrictions

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)

Example

To add ports et.1.1, et.1.2, and et.1.3 to SmartTRUNK st.1:

```
ssr(config)# smarttrunk add ports et.1.(1-3) to st.1
```

smarttrunk clear load-distribution

Purpose

Clears load distribution statistics for ports in a SmartTRUNK.

Format

smarttrunk clear load-distribution <*smarttrunk list*> | **all-smarttrunks**

Mode

Enable

Description

The **smarttrunk clear load-distribution** command is used in conjunction with the **smarttrunk show distribution** command, which gathers statistics for the transmitted bytes per second flowing through the SmartTRUNK and each port in it. The **smarttrunk clear load-distribution** command lets you reset load distribution statistics to zero.

Parameters

<*smarttrunk list* >	Is the name of one or more existing SmartTRUNKs.
all-smarttrunks	Causes load distribution information to be cleared for all SmartTRUNKs.

Restrictions

None.

Example

To clear load distribution information from SmartTRUNK st.1:

```
ssr# smarttrunk clear load-distribution st.1
```

smarttrunk create

Purpose

Creates a SmartTRUNK and specifies a control protocol for it.

Format

smarttrunk create <*smarttrunk*> **protocol no-protocol | huntgroup**

Mode

Configure

Description

The **smarttrunk create** command allows you to create a SmartTRUNK logical port. Once you have created a SmartTRUNK port, you add physical ports to it with the **smarttrunk add ports** command.

SmartTRUNKs on the SSR are compatible with the DEC Hunt Groups control protocol. If you are connecting the SmartTRUNK to another SSR, Cabletron switch, or Digital GIGAswitch/Router, you can specify that the SmartTRUNK use this control protocol. SmartTRUNKing and Hunt Groups are comprised of two protocols:

- Logical Link Aging Protocol (LLAP) – Assists in learning and aging
- Physical Link Affinity Protocol (PLAP) – Monitors and maintains the trunking states

SmartTRUNKs are also compatible with devices that do not support the Hunt Groups control protocol, such as those that support Cisco's EtherChannel technology. If you are connecting a SmartTRUNK to devices that do not support Hunt Groups, no control protocol is used. You must specify the **no-protocol** keyword in the **smarttrunk create** command.

Parameters

<*smarttrunk*>	Is the name of the SmartTRUNK to create. The name of the SmartTRUNK must be in the form st.*x*; for example, st.1.
no-protocol	Specifies that no control protocol be used. Use this keyword if the SmartTRUNK is connected to a device that does not support the DEC

Hunt Group control protocol (that is, a device from a vendor other than Cabletron or DIGITAL).

huntgroup Specifies that the DEC Hunt Group control protocol be used. Use this keyword if you are connecting the SmartTRUNK to another SSR, Cabletron switch, or Digital GIGAswitch/Router.

Restrictions

None.

Example

The following command creates a SmartTRUNK named st.1, using the DEC Hunt Group control protocol.

```
ssr(config)# smarttrunk create st.1 protocol huntgroup
```

smarttrunk set load-policy

Purpose

Specifies how traffic is distributed across the ports in a SmartTRUNK.

Format

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

Mode

Configure

Description

The **smarttrunk set load-policy** command lets you specify how a SmartTRUNK distributes traffic among its ports. There are two options: **round-robin** (the default) and **link-utilization**.

Round-robin means that flows are assigned to ports on a sequential basis. The first flow goes to the first port in the SmartTRUNK, the second flow to the second port, and so on. Link-utilization means that a flow is assigned to the least-used port in the SmartTRUNK.

Parameters

<*smarttrunk list* >	Is the name of one or more SmartTRUNKs.
all-smarttrunks	Specifies that the command be applied to all SmartTRUNKs.
round-robin	Specifies that traffic be distributed evenly across all ports.
link-utilization	Specifies that packets should be sent to the least-used port in the SmartTRUNK.

Restrictions

None.

Example

To specify that SmartTRUNK st.1 distribute flows sequentially among its component ports:

```
ssr(config)# smarttrunk set load-policy on st.1 round-robin
```

smarttrunk show

Purpose

Displays information about SmartTRUNKs on the SSR

Format

smarttrunk show trunks

smarttrunk show distribution | protocol-state | connections <*smarttrunk list*> **| all-smarttrunks**

Mode

Enable

Description

The **smarttrunk show** command shows statistics about SmartTRUNKs on the SSR.

Parameters

trunks	Shows information about all SmartTRUNKs, including active and inactive ports, and the control protocol used.
distribution	Provides statistics on how traffic is distributed across the ports in a SmartTRUNK.
protocol-state	Shows information about the control protocol on a SmartTRUNK.
connections	Shows information about the SmartTRUNK connection, including the MAC address of the remote switch, and the module number and port number of each remote port. Connection information is reported only if the Hunt Group protocol is enabled for the SmartTRUNK.
<*smarttrunk list* >	Is the name of one or more SmartTRUNKs.
all-smarttrunks	Specifies that the command be applied to all SmartTRUNKs.

Restrictions

None.

Examples

To display information about all SmartTRUNKs on the SSR:

```
ssr# smarttrunk show trunks

Flags: D - Disabled  I - Inactive

SmartTRUNK Active Ports        Inactive Ports      Primary Port  Protocol  Load-Policy    Flags
---------- ------------        --------------      ------------  --------  -----------    -----
st.1                           et.3.(7-8)           None          None      RR
```

To show how traffic is distributed across the ports on SmartTRUNK st.1:

```
ssr# smarttrunk show distribution st.1

SmartTRUNK  Member Port  Total (bytes/sec)    Port (bytes/sec)     % Load
----------  -----------  -----------------    ----------------     ------
st.1        et.2.4       7660268              2872592              37
st.1        et.2.5       7660268              1915084              25
st.1        et.2.6       7660268              2872592              37
```

To show information about the control protocol for SmartTRUNK st.1:

```
ssr# smarttrunk show protocol-state st.1

SmartTRUNK  Protocol        State     Port    Port State
----------  --------        -----     ----    ----------
st.1        HuntGroup       Down      et.3.1  Negotiate
                                      et.3.2  Negotiate
```

To show connection information for all SmartTRUNKs:

```
ssr# smarttrunk show connections all-smarttrunks
SmartTRUNK Local Port  Remote Switch        Remote Module   Remote Port  State
---------- ----------  -------------        -------------   -----------  -----
st.1       et.2.1      Cabletron A9:6E:57   3                 1            Up
st.1       et.2.2      Cabletron A9:6E:57   3                 2            Up
st.1       et.2.3      Cabletron A9:6E:57   3                 3            Up
st.1       gi.3.1      Cabletron A9:6E:57   4                 5            Up
st.2       et.2.4      --                   --              --           Up
st.2       et.2.5      --                   --              --           Up
st.2       et.2.6      --                   --              --           Up
```

Note: In the example above, SmartTRUNK st.2 has no control protocol enabled, so no connection information is reported.

Chapter 53

snmp Commands

The SNMP commands let you set and show SNMP parameters including SNMP community names and IP host targets for SNMP traps.

Command Summary

Table 38 lists the **snmp** commands. The sections following the table describe the command syntax.

Table 38. snmp Commands

snmp disable trap authentication | link-up-down
snmp set chassis-id <*chassis-name*>
snmp set community <*community-name*> privilege **read | read-write**
snmp set target <*IP-addr*> **community** <*community-name*> **[status enable | disable]**
snmp show access | all | chassis-id | community | statistics | trap
snmp stop

snmp disable trap

Purpose

Disable specific SNMP trap types.

Format

snmp disable trap authentication | link-up-down

Mode

Configure

Description

The **snmp disable trap** command controls the types of traps the SSR emits based trap type. You can disable the following trap types:

- Authentication – use the **authentication** keyword to prevent the SSR from sending a trap each time it receives an invalid community string or invalid Telnet password.
- Link-state change – use the **link-up-down** keyword to prevent the SSR from sending a trap each time a port changes operational state.

Parameters

authentication Disables authentication traps, which the SSR sends when it receives an invalid SNMP community string or Telnet password.

link-up-down Disables link-state change traps, which the SSR sends when a port's operational state changes.

Restrictions

None.

snmp set chassis-id

Purpose

Set the SSR's chassis ID using SNMP.

Format

snmp set chassis-id <*chassis-name*>

Mode

Configure

Description

The **snmp set chassis-id** command lets you set a string to give the SSR an SNMP identity.

Parameters

<*chassis-name*> Is a string describing the SSR.

Restrictions

None.

snmp set community

Purpose

Set an SNMP community string and specify the access privileges for that string.

Format

snmp set community <*community-name*> **privilege read | read-write**

Mode

Configure

Description

The **snmp set community** command sets a community string for SNMP access to the SSR. SNMP management stations that want to access the SSR must supply a community string that is set on the switch. This command also sets the level of access to the SSR to read-only or read-write. Communities that are read-only allow SNMP GETs but not SNMP SETs. Communities that have read-write access allow both SNMP GETs and SNMP SETs.

Parameters

community <*community-name*>
Character string for the community string.

privilege read | read-write
Access level. Specify one of the following:

read	Allows SNMP GETs but not SNMP SETs.
read-write	Allows SNMP GETs and not SNMP SETs.

Restrictions

None.

Example

To set the SNMP community string to "public," which has read-only access:

```
ssr(config)# snmp set community public privilege read
```

snmp set target

Purpose

Sets the target IP address and community string for SNMP traps.

Format

snmp set target <*IP-addr*> **community** <*community-name*> **[status enable | disable]**

Mode

Configure

Description

The **snmp set target** command specifies the IP address of the target server to which you want the SSR to send SNMP traps. Trap targets ar enabled by default but you can use the status argument to disable or re-enable a target.

Note: In general, community strings sent with traps should not have read-write privileges.

Parameters

<IP-addr>
Is the IP address of the management station from which you want to be able to access the traps.

Note: The target IP address should be locally attached to the SSR. Cold start traps might not reach their destination if the target requires dynamic route table entries to be forwarded correctly. The SSR will retry every minute up to four minutes on the cold-start trap.

<community-name>
Is the name of the SNMP community for which you are setting the trap target.

status enable | disable
Re-enables or disables the target.

Restrictions

None.

snmp show

Purpose

Shows SNMP information.

Format

snmp show access | all | chassis-id | community | statistics | trap

Mode

Enable

Description

The **snmp show** command shows the following SNMP information:

- Community strings set on the SSR
- SNMP Statistics
- IP address of SNMP trap target server

Parameters

access	Displays the last five SNMP clients to access the SSR.
all	Displays all SNMP information (equivalent to specifying all the other keywords).
chassis-id	Displays the SSR's SNMP name.
community	Displays the SSR's community string.
statistics	Displays SNMP statistics.
trap	Displays the IP address of the trap target server.

Restrictions

None.

Examples

The following command displays a log of SNMP access to the SSR. The host that accessed the SSR and the SSR system time when the access occurred are listed.

```
ssr(confg)# snmp show access
SNMP Last 5 Clients:
      10.15.1.2      Wed Feb 10 18:42:59 1999
      10.15.1.2      Wed Feb 10 18:42:55 1999
      10.15.1.2      Wed Feb 10 18:42:56 1999
      10.15.1.2      Wed Feb 10 18:42:57 1999
      10.15.1.2      Wed Feb 10 18:42:58 1999
```

To display the SNMP identity of the SSR:

```
ssr(confg)# snmp show chassis-id

SNMP Chassis Identity:
s/n 123456
```

To display the IP address of the trap target server:

```
ssr(confg)# snmp show trap

Trap Table:
Index  Trap   Target Addr   Community String   Status
1.            10.15.1.2     public             enabled
2.            1.2.3.4       public123          disabled
3.            5.6.7.8       public20           disabled
```

snmp stop

Purpose

Stop SNMP access to the device.

Format

snmp stop

Mode

Configure

Description

The **snmp stop** command stops SNMP access to the SSR. The SSR will still finish all active requests but will then disregard future requests. When you issue this command, UDP port 161 is closed.

Parameters

None.

Restrictions

None.

Chapter 54

statistics Commands

The **statistics** commands let you display statistics for various SSR features. You also can clear some statistics.

Command Summary

Table 39 lists the statistics commands. The sections following the table describe the command syntax.

Table 39. statistics commands

statistics clear port-errors | **port-stats** | **rmon** <*port-list*>
statistics show <*statistic-type*> **[**<*port-list*>**]** **Note:** Not all statistic types accept a port list.

statistics clear

Purpose

Clear statistics.

Format

statistics clear <*statistic-type*> <*port-list*>

Mode

Enable

Description

The **statistics clear** command clears port statistics, error statistics, or RMON statistics. When you clear statistics, the SSR sets the counters for the cleared statistics to 0, then begins accumulating the statistics again.

Parameters

<*statistic-type*>
Type of statistics you want to clear. Specify one of the following:

port-errors Clears all error statistics for the specified port.

port-stats Clears all normal (non-error) statistics for the specified port.

rmon Clears all RMON statistics for the specified port.

<*port-list*>
The ports for which you are clearing statistics. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8). Specify **all-ports** to clear statistics for all the SSR ports.

Restrictions

None.

statistics show

Purpose

Display statistics.

Format

statistics show <*statistic-type*> <*port-list*>

Mode

Enable

Parameters

<*statistic-type*>
The type of statistics you want to display. Specify one of the following. Some statistics options apply system-wide, while others apply only to the Control Module.

System-wide statistics:

port-errors Shows error statistics for ports.

port-stats Shows normal (non-error) port statistics.

rmon Shows RMON statistics.

rarp Shows Reverse Address Resolution Protocol (RARP) statistics.

top Shows the most active tasks. Task usage is shown as both a percentage of total CPU utilization and a percentage of other tasks running on the system.

ip-interface <*options*> Shows IP interface statistics.

ipx-interface <*options*>Shows IPX interface statistics.

For **ip-interface** and **ipx-interface**, the interface name, input and output frames, and input and output errors are displayed. However, you can use one or more of the following <*options*> to control the type of information displayed:

packets Displays packet statistics.

bytes Displays byte statistics.

errors Displays error statistics.

input If specified following one of the three options listed above, displays only input statistics for that option. Both input and output statistics are displayed by default.

output If specified following one of the three options listed above, displays only output statistics for that option.

verbose Displays all statistics.

Control-Module statistics:

icmp Shows ICMP statistics.

ip Shows IP statistics.

ip-routing Shows IP unicast routing statistics.

ipx Shows IPX statistics.

ipx-routing Shows IPX unicast routing statistics.

multicast Shows IP multicast statistics.

tcp Shows TCP statistics.

udp Shows UDP statistics.

<port-list>
For system-wide statistics options, the ports for which you are showing statistics. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8). Specify **all-ports** to show statistics for all the SSR ports.

Restrictions

None.

Chapter 55

stp Commands

The stp commands let you display and change settings for the default Spanning Tree.

Command Summary

Table 40 lists the stp commands. The sections following the table describe the command syntax.

Table 40. stp commands

stp enable port <*port-list*>
stp set bridging [forward-delay <*num*>**] [hello-time** <*num*>**] [max-age** <*num*>**] [priority** <*num*>**]**
stp set port <*port-list*> **priority** <*num*> **port-cost** <*num*>
stp show bridging-info

stp enable port

Purpose

Enable STP on one or more ports.

Format

stp enable port <*port-list*>

Mode

Configure

Description

The **stp enable port** command enables STP on the specified ports.

Parameters

<*port-list*> The ports on which you are enabling STP. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

Restrictions

None

stp set bridging

Purpose

Set STP bridging parameters.

Format

stp set bridging [forward-delay <*num*>**] [hello-time** <*num*>**] [max-age** <*num*>**] [priority** <*num*>**]**

Mode

Configure

Description

The **stp set bridging** command lets you configure the following STP parameters:

- Bridging priority
- Hello time
- Maximum age
- Forward delay

Parameters

forward-delay <*num*>
Sets the STP forward delay for the SSR. The forward delay is measured in seconds. Specify a number from 4– 30. The default is 15.

hello-time <*num*>
Sets the STP hello time for the SSR. The hello time is measured in seconds. Specify a number from 1– 10. The default is 2.

max-age <*num*>
Sets the STP maximum age for the SSR. Specify a number from 6–40. The default is 20.

priority <*num*>
Sets the STP bridging priority for the SSR. Specify a number from 0 – 65535. The default is 32768

Restrictions

None.

Examples

To set the bridging priority of Spanning Tree for the entire SSR to 1:

```
ssr(config)# stp set bridging priority 1
```

stp set port

Purpose

Set STP port priority and port cost for ports.

Format

stp set port <*port-list*> **priority** <*num*> **port-cost** <*num*>

Mode

Configure

Description

The **stp set port** command sets the STP priority and port cost for individual ports.

Parameters

port <*port-list*>
The port(s) for which you are setting STP parameters. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

priority <*num*>
The priority you are assigning to the port(s). Specify a number from 0– 255. The default is 128.

port-cost <*num*>
The STP cost you are assigning to the port(s). Specify a number from 1– 65535. The default depends on the port speed: 1 for Gigabit (100-Mbps) ports, 10 for 100-Mbps ports, and 100 for 10-Mbps ports.

Restrictions

None.

stp show bridging-info

Purpose

Display STP bridging information.

Format

stp show bridging-info

Mode

Enable

Description

The **stp show bridging-info** command displays STP bridging information for the SSR.

Parameters

None.

Restrictions

None.

Chapter 56

system Commands

The **system** commands let you display and change system parameters.

Command Summary

Table 41 lists the system commands. The sections following the table describe the command syntax.

Table 41. system commands

system hotswap out | **in channel** <*number*>
system image add <*IPaddr-or-hostname*> <*filename*>
system image choose <*filename*>
system image list
system image delete <*filename*>
system kill telnet-session <*session-id*>
system promimage upgrade <*hostname-or-IPaddr*> <*filename*>
system set bootprom netaddr <*IPaddr*> **netmask** <*IPnetmask*> **tftp-server** <*IPaddr*> **[tftp-gateway** <*IPaddr*>**]**
system set contact <*system-contact*>
system set date year <*year*> **month** <*month*> **day** <*day*> **hour** <*hour*> **min** <*min*> **second** <*sec*>
system set daylight-savings

Table 41. system commands (Continued)

system set dns server <*IPaddr*>**[,**<*IPaddr*>**[,**<*IPaddr*>**]]** domain <*name*>
system set location <*location*>
system set login-banner <*string*> **| none**
system set name <*system-name*>
system set password <*mode*> <*string*>	none
system set poweron-selftest [on	quick]
system set show-config alphabetical
system set syslog [server <*hostname-or-IPaddr*>**] [level** <*level-type*>**]** **[facility** <*facility-type*>**] [buffer-size** <*size*>**] [source <***source-IPaddr***>]**
system set terminal baud <*baud-rate*>	**columns** <*num*>	**rows** <*num*>
system set timezone <*timezone*>	<*minutes*>
system show <*system-parm*>

system hotswap

Purpose

Activates or deactivates a line card.

Format

system hotswap out | in slot <*number*>

Mode

Enable

Description

The **system hotswap out** command deactivates a line card in a specified slot on the SSR, causing it to go offline. The command performs the same function as if you had pressed the Hot Swap button on the line card.

The **system hotswap in** command causes a line card that was deactivated with the **system hotswap out** command to go online again. The command performs the same function as if you had removed the card from its slot and inserted it again.

See the *SmartSwitch Router User Reference Manual* for more information on hot swapping line cards.

Parameters

out
Causes the line card in the specified slot to be deactivated.

in
Causes an inactive line card in the specified slot to be reactivated.

Note: The **system hotswap in** command works only on a line card that was deactivated with the **system hotswap out** command.

slot <*number*>
Is the slot where the line card resides. Specify 1-7 for the SSR 8000 or 1-15 for the SSR 8600.

Restrictions

None.

Example

To deactivate the line card in slot 7 on the SSR:

```
ssr# system hotswap out slot 7
```

system image add

Purpose

Copy a system software image to the SSR.

Format

system image add <*IPaddr-or-hostname*> <*filename*> **[primary-cm] [backup-cm]**

Mode

Enable

Description

The **system image add** command copies a system software image from a TFTP server into the PCMCIA flash card on the Control Module. By default, if the SSR has two Control Modules, the system software image is copied to both Control Modules.

Parameters

<*IPaddr-or-hostname*>
Is the IP address or host name of the TFTP server or a TFTP URL.

<*filename*>
Is the name of the system software image file.

primary-cm
Copies the system software image only to the primary Control Module.

backup-cm
Copies the system software image only to the secondary Control Module.

Restrictions

None.

Example

To download the software image file named `img.tar.gz` from the TFTP server 10.1.2.3:

```
ssr# system image add tftp://10.1.2.3/images/img.tar.gz
```

system image choose

Purpose

Select a system software image file.

Format

system image choose <*filename*>

Mode

Enable

Description

The **system image choose** command specifies the system software image file on the PCMCIA flash card that you want the SSR to use the next time you reboot the system.

Parameters

<*filename*> The name of the system software image file.

Restrictions

None.

system image delete

Purpose

Deletes a system software image file from the PCMCIA flash card.

Format

system image delete <*filename*>

Mode

Enable

Description

The **system image delete** command deletes a system software image file from the PCMCIA flash card on the Control Module.

Parameters

<*filename*> The name of the system software image file you want to delete.

Restrictions

None.

system image list

Purpose

Lists the system software image files on the PCMCIA flash card.

Format

system image list

Mode

Enable

Description

The **system image list** command lists the system software image files contained on the PCMCIA flash card on the Control Module.

Parameters

None.

Restrictions

None.

system kill telnet-session

Purpose

Kills a specified Telnet session.

Format

system kill telnet-session <*session-id*>

Mode

Enable

Description

The **system kill telnet-session** command kills the Telnet session specified by the session ID. Use the **system show users** command to display the list of current Telnet users and session IDs.

Parameters

<*session-id*>
The Telnet connection slot number, which can be 0, 1, 2, or 3. The **system show users** command displays the session ID number in the first column. You can only specify one session ID per **system kill telnet-session** command.

Restrictions

None.

Example

To show the active Telnet sessions.

```
ssr# system show users
Current Terminal User List:
# Login ID       Mode           From              Login Timestamp
- --------       ----           ----              ---------------
                enabled        console           Thu Feb 25 13:07:411999
0               enabled        10.9.0.1          Thu Feb 25 13:07:591999
2                login-prompt   10.9.0.1
3                login-prompt   10.9.0.1
```

Then, to kill Telnet session 2:

```
ssr# system kill telnet-session 2
Telnet session 2 (from 10.9.0.1) killed
```

system promimage upgrade

Purpose

Upgrades the boot PROM software on the Control Module.

Format

system promimage upgrade <*IPaddr-or-hostname*> <*filename*>

Mode

Enable

Description

The **system promimage upgrade** command copies and installs a boot PROM software image from a TFTP server onto the internal memory on the Control Module. The boot PROM software image is loaded when you power on the SSR and in turn loads the system software image file.

Parameters

<*IPaddr-or-hostname*>
The IP address or host name of the TFTP server or a TFTP URL.

<*filename*>
The name of the boot PROM software image file.

Restrictions

None.

Example

The command in the following example downloads a boot PROM image file from the TFTP server 10.50.89.88.

```
ssr# system promimage upgrade tftp://10.50.89.88/qa/prom-upgrade
Downloading image ‘qa/prom-upgrade’ from host ‘10.50.89.88’
tftp complete
checksum valid. Ready to program.
flash found at 0xbfc00000
erasing...
programming...
verifying...
programming successful.
Programming complete.
```

system set bootprom

Purpose

Sets parameters for the boot PROM.

Format

system set bootprom netaddr <*IPaddr*> **netmask** <*IPnetmask*>
tftp-server <*IPaddr*> **[tftp-gateway** <*Ipaddr*>**]**

Mode

Configure

Description

The **system set bootprom** command sets parameters to aid in booting the SSR's system software image remotely over the network. You can use this command to set the SSR's IP address, subnet mask, TFTP boot server address, and gateway address.

Note: These parameters apply only to the Control Module's en0 Ethernet interface.

Parameters

netaddr <*IPaddr*>
The IP address the SSR uses during the boot exchange with the TFTP boot server.

netmask <*IPnetmask*>
The subnet mask the SSR uses during the boot exchange.

tftp-server <*IPaddr*>
The TFTP boot server's IP address.

tftp-gateway <*Ipaddr*>
The gateway that connects the SSR to the TFTP boot server.

Restrictions

None.

Example

The command in the following example configures the SSR to use IP address 10.50.88.2 to boot over the network from TFTP boot server 10.50.89.88.

```
ssr(config)# system set bootprom netaddr 10.50.88.2 netmask 255.255.0.0
    tftp-server 10.50.89.88
```

system set contact

Purpose

Set the contact name and information for this SSR.

Format

system set contact <*system-contact*>

Mode

Configure

Description

The **system set contact** command sets the name and contact information for the network administrator responsible for this SSR.

Parameters

<*system-contact*>
A string listing the name and contact information for the network administrator responsible for this SSR. If the string contains blanks or commas, you must use the quotation marks around the string. (Example: **"Jane Doe, janed@corp.com, 408-555-5555 ext. 555"**.)

Restrictions

None.

system set date

Purpose

Set the system time and date.

Format

system set date year <*year*> **month** <*month*> **day** <*day*>
hour <*hour*> **min** <*min*> **second** <*sec*>

Mode

Enable

Description

The **system set date** command sets the system time and date for the SSR. The SSR keeps the time in a battery-backed realtime clock. To display the time and date, enter the **system show date** command.

Parameters

year <*number*>
Four-digit number for the year. (Example: **1998**)

month <*month-name*>
Name of the month. You must spell out the month name. (Example: **March**)

day <*day*>
Number from 1 – 31 for the day.

hour <*hour*>
Number from 0 – 23 for the hour. (The number **0** means midnight.)

minute <*minute*>
Number from 0 – 59 for the hour.

second <*second*>
Number from 0 – 59 for the second.

Restrictions

None.

system set daylight-saving

Purpose

Enable daylight saving for the local time zone.

Format

system set daylight-saving

Mode

Configure

Description

If daylight savings time is in effect in the local time zone, use the **system set daylight-saving** command to enable it on the SSR. When daylight savings time is in effect, an additional hour is subtracted from your UCT offset. This command may be required if you use NTP (Network Time Protocol) to synchronize the system's real time clock. To disable daylight savings time on the SSR negate this command.

Parameters

None.

Restrictions

None.

Example

When daylight savings time begins in the local time zone, enable it on the SSR with the following command:

```
ssr(config)# system set daylight-saving
```

When daylight savings time ends in the local time zone, disable it on the SSR with the following command:

```
ssr(config)# no system set daylight-saving
```

system set dns

Purpose

Configure the SSR to reach up to three DNS servers.

Format

system set dns server ["]<IPaddr> [<*IPaddr*>] [<*IPaddr*>]["] **domain** <*name*>

Mode

Configure

Description

The **system set dns** command configures the SSR to reach up to three DNS servers. You also can specify the domain name to use for each DNS query by SSR.

Parameters

["]<*IPaddr*> [<*IPaddr*>] [<*IPaddr*>]["]
IP address of the DNS server. Specify the address in dotted-decimal notation. You can specify up to three DNS servers separated by single spaces in the command line.

Note: If you specify more than one IP address, you must surround the IP address specification with a set of quotes.

<*domain-name*>
Domain name for which the server is an authority.

Restrictions

None.

Examples

To configure a single DNS server and configure the SSR's DNS domain name to "mrb.com":

```
ssr(config)# system set dns server 10.1.2.3 domain mrb.com
```

To configure three DNS servers and configure the SSR's DNS domain name to "mrb.com":

```
ssr(config)# system set dns server "10.1.2.3 10.2.10.12 10.3.4.5" domain
    mrb.com
```

system set location

Purpose

Set the system location.

Format

system set location <*location*>

Mode

Configure

Description

The **system set location** command adds a string describing the location of the SSR. The system name and location can be accessed by SNMP managers.

Parameters

<*location*> A string describing the location of the SSR. If the string contains blanks or commas, you must use quotation marks around the string. (Example: **"Bldg C, network control room"**.)

Restrictions

None.

system set login-banner

Purpose

Set the system login banner.

Format

system set login-banner <*string*> | **none**

Mode

Configure

Description

The **system set login-banner** command configures the initial login banner that one sees when logging into the SSR. The banner may span multiple lines by adding line-feed characters in the string, "**\n**".

Parameters

<*string*> Is the text of the login banner for the SSR. The banner may span multiple lines by having line-feed characters in the string, "**\n**".

none Specifies that no login-banner be used on the SSR.

Restrictions

None.

Example

The following example configures a multi-line login banner:

```
ssr(config)# system set login-banner "Server network SSR\nUnauthorized
Access Prohibited"
```

The next person to log into the SSR would see the following:

```
Server network SSR
Unauthorized Access Prohibited

Press RETURN to activate console...
```

If you do not want any login-banner at all, enter the following:

```
ssr(config)# system set login-banner none
```

system set name

Purpose

Set the system name.

Format

system set name <*system-name*>

Mode

Configure

Description

The **system set name** command configures the name of the SSR. The SSR name will use the name as part of the command prompt.

Parameters

<*system-name*> The hostname of the SSR. If the string contains blanks or commas, you must use quotation marks around the string. (Example: **"Mega-Corp** SSR **#27"**.)

Restrictions

None.

system set password

Purpose

Set passwords for various CLI access modes.

Format

system set password <*mode*> <*string*> | **none**

Mode

Configure

Description

The **system set password** command sets or changes the passwords for the Login and Enable access modes.

Note: If a password is configured for the Enable mode, the SSR prompts for the password when you enter the **enable** command. Otherwise, the SSR displays a message advising you to configure an Enable password, then enters the Enable mode. From the Enable mode, you can access the Configure mode to make configuration changes.

Parameters

<*mode*>
The access mode for which you are setting a password. Specify one of the following:

login The password required to start a CLI session. The SSR prompts for this password when the system finishes booting.

enable The password for entering the Enable mode.

<*string*> | **none**
The password. If you specify **none**, no password is required.

Note: You cannot use the string "none" as a password.

Restrictions

The SSR stores passwords in the Startup configuration file. If you copy a configuration file from one SSR to another, the passwords in the file also are copied and will be required on the new SSR.

When you activate a new password by copying the password set command to the active configuration, the SSR replaces the command with a **system set hashed-password** command, which hides the password text in the configuration file so that the password is not visible to others if they examine the configuration file.

To remove a password, enter the following command while in Configure mode:

```
ssr(config)# system set password <mode> none
```

system set poweron-selftest

Purpose

Specify the type of Power-On-Self-Test (POST) to perform during system bootup.

Format

system set poweron-selftest [on | quick]

Mode

Configure

Description

The **system set poweron-selftest** command configures the type of Power-On-Self-Test (POST) the SSR should perform during the next system bootup. By default, no POST is performed during system bootup. To perform POST, you must use this command to specify which type of test to run, **quick** or **full**. Once POST enabled, to turn off POST, you simply negate this command (using the **negate** command).

Parameters

on The SSR will perform a **full** test during the next system bootup.

quick The SSR will perform a **quick** test during the next system bootup.

Restrictions

None.

system set show-config

Purpose

Specify how configuration commands should be displayed.

Format

system set show-config alphabetical

Mode

Configure

Description

The **show** and **system show active-config** commands normally display the configuration commands in the order that they are executed. The **system set show-config** command changes the way the configuration commands are shown.

Parameters

alphabetical Shows the configuration commands in alphabetical order.

Restrictions

None.

Example

To display the configuration commands in alphabetical order:

```
ssr(config)# system set show-config alphabetical
```

system set syslog

Purpose

Identify a Syslog server to which the SSR can send Syslog messages

Format

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

Mode

Configure

Description

The **system set syslog** command identifies the Syslog server to which the SSR should send system messages. You can control the type of messages to send as well as the facility under which the message is sent. The type of messages to send is based on the severity of the message (controlled by the option **level**). Messages can also be sent under a specific facility. There are 11 facilities supported by the SSR. On the Syslog server, you can decide what to do with these messages based on the level as well as the facility. For example, you might choose to discard the messages, write them to a file or send them out to the console. You can further identify the source of the system messages sent to the Syslog server by specifying a source IP address for the Syslog on the SSR.

The SSR keeps the last <*n*> messages in a local circular buffer. By default, this buffer keeps the last 10 Syslog messages. You can change the buffer size to hold anywhere from 10 – 50 messages. To view the current buffer size, enter the **system show syslog buffer** command.

Parameters

<hostname-or-IP-addr>
Hostname or IP address of the SYSLOG server.

<level-type>
Level of messages you want the SSR to log. Specify one of the following:

fatal Logs only fatal messages.

error Logs fatal messages and error messages.

warning Logs fatal messages, error messages, and warning messages. This is the default.

info Logs all messages, including informational messages.

<facility-type>

Type of facility under which you want messages to be sent. By default, unless specified otherwise, messages are sent under facility *local7*. The facility-type can be one of the following:

kern kernel messages

user user messages

daemon daemon messages

local0 Reserved for local use

local1 Reserved for local use

local2 Reserved for local use

local3 Reserved for local use

local4 Reserved for local use

local5 Reserved for local use

local6 Reserved for local use

local7 Reserved for local use

<source-IPaddr>

Source IP address of the messages sent to the Syslog server. You must specify a Unicast IP address in the form a.b.c.d.

<size>

The Syslog message buffer size. The size specifies how many messages the Syslog buffer can hold. You can specify a number from 10 – 50, giving the buffer a capacity to hold from 10– 50 Syslog messages. The default is 10.

Restrictions

None.

Example

To log only fatal and error level messages to the syslog server on 10.1.43.77:

```
ssr(config)# system set syslog server 10.1.43.77 level error
```

system set terminal

Purpose

Sets global terminal parameters.

Format

system set terminal baud <*baud-rate*> | **columns** <*num*> | **rows** <*num*>

Mode

Configure

Description

The **system set terminal** command globally sets parameters for a serial console's baud rate, output columns, and output rows.

Parameters

baud <*baud-rate*>
Sets the baud rate. You can specify one of the following:

- **300**
- **600**
- **1200**
- **2400**
- **4800**
- **9600**
- **19200**
- **38400**

columns <*num*>
Sets the number of columns displayed at one time.

rows <*num*>
Sets the number of rows displayed at one time.

Restrictions

None.

Example

The command in the following example sets the baud rate, number of columns, and number of rows for the management terminal connected to the System Control module.

```
ssr(config)# system set terminal baud 38400 columns 132 rows 50
```

system set timezone

Purpose

Sets time zone information or time offset.

Format

system set timezone <*timezone*> | <*minutes*>

Mode

Configure

Description

The **system set timezone** command sets the local time zone for the SSR. You can use one of the time zone keywords to specify the local time zone or specify the time offset in minutes. You must configure the time zone in order to use NTP (Network Time Protocol) to synchronize the SSR's real time clock.

Parameters

<*timezone*>

Sets the time zone using one of the following keywrods:

est	Eastern Standard Time (UCT -05:00)
cst	Central Standard Time (UCT -06:00)
mst	Mountain Standard Time (UCT -07:00)
pst	Pacific Standard Time (UCT -08:00)
uct-12	Eniwetok, Kawajalein (UCT -12:00)
uct-11	Midway Island, Samoa (UCT -11:00)
uct-10	Hawaii (UCT -10:00)
uct-9	Alasaka (UCT -09:00)
uct-8	Pacific Standard Time (UCT -08:00)
uct-7	Mountain Standard Time (UCT -07:00)

uct-6	Central Standard Time (UCT -06:00)
uct-5	Eastern Standard Time (UCT -05:00)
uct-4	Caracas, La Paz (UCT -04:00)
uct-3	Buenos Aires, Georgetown (UCT -03:00)
uct-2	Mid-Atlantic (UCT -02:00)
uct-1	Azores, Cape Verde Island (UCT -01:00)
uct	Greenwich, London, Dublin (UCT)
uct+1	Berlin, Madrid, Paris (UCT +01:00)
uct+2	Athens, Helsinki, Istanbul, Cairo (UCT +02:00)
uct+3	Moscow, Nairobi, Riyadh (UCT +03:00)
uct+4	Abu Dhabi, Kabul(UCT +05:00)
uct+5	Pakistan (UCT +05:00)
uct+5:30	India (UCT +05:30)
uct+6	Bangladesh (UCT +06:00)
uct+7	Bangkok, Jakarta (UCT +07:00)
uct+8	Beijing, Hong Kong, Singapore(UCT +08:00)
uct+9	Japan, Korea (UCT +09:00)
uct+10	Sydney, Guam (UCT +10:00)
uct+11	Solomon Is. (UCT +11:00)
uct+12	Fiji, Marshall Is. Auckland (UCT +12:00)

<minutes>
Specify the time zone offset in minutes. Valid values are between -720 minutes to + 720 minutes.

Restrictions

None.

Example

To set the local time zone to Pacific Standard Time (UCT -8:00).

```
ssr(config)# system set timezone pst
```

system show

Purpose

Show system information.

Format

system show <*system-parm*>

Mode

Enable

Description

The **system show command** shows the active settings for the following system parameters:

- Active configuration (CLI configuration of the running system)
- Size of the Syslog message buffer
- Contact information for the SSR administrator (if you set one using the **system set contact** command)
- Current system time and date (if you set them using **system set date** command)
- Time that has elapsed since the SSR was rebooted and the system time and date when the last reboot occurred
- IP address(es) and domain name of DNS servers the SSR can use (if you set them using **system set dns** command)
- Hardware information
- Location of the SSR (if you set one using the **system set location** command)
- System name of the SSR (if you set one using the **system set name** command)
- IP address or hostname of SYSLOG server and the message level (if you set these parameters using the **system set syslog** command)
- Configuration changes in the scratchpad that are waiting for activation
- Software version running on the Control Module
- Last five Telnet connections to the SSR

- Current Telnet sessions on the SSR
- CPU and other resource usage

Parameters

<system-parm>
System parameter you want to display. Specify one of the following:

active-config
Shows the active configuration of the system.

bootlog
Shows the contents of the boot log file, which contains all the system messages generated during bootup.

bootprom
Shows boot PROM parameters for TFTP downloading of the system image. This information is useful only if you have configured the system to download the system image via TFTP.

capacity all | chassis | task | cpu | memory
Shows usage information about various resources on the **SSR**.

contact
Shows the contact information (administrator name, phone number, and so on).

cpu-utilization
Shows the percentage of the CPU that is currently being used.

date
Shows the system time and date.

dns
Shows the IP addresses and domain names for the DNS servers the SSR can use.

environmental
Shows environmental information, such as temperature and power supply status.

hardware
Shows hardware information.

location
Shows the SSR's location.

login-banner
Shows the SSR's login banner. The login banner can be configured using the **system set login-banner** command.

name
Shows the SSR's name.

poweron-selftest-mode
Shows the type of Power-On Self Test (POST) that should be performed, if any.

scratchpad
Shows the configuration changes in the scratchpad. These changes have not yet been activated.

startup-config
Shows the contents of the Startup configuration file.

switching-fabric
Shows the status of the switching fabric module. This command is valid only for the SSR 8600.

syslog
Shows the IP address of the SYSLOG server and the level of messages the SSR sends to the server.

syslog buffer
Shows how many Syslog messages the SSR's Syslog message buffer can hold.

telnet-access
Lists the last five Telnet connections to the SSR.

terminal
Shows the default terminal settings (number of rows, number of columns, and baud rate).

timezone
Shows the time zone offset from UCT in minutes.

uptime
Show how much time has elapsed time since the most recent reboot.

users
Shows the current Telnet connections to the SSR.

version
Shows the software version running on the SSR.

Restrictions

None.

Chapter 57

tacacs Commands

The **tacacs** commands let you secure access to the SSR using the Terminal Access Controller Access Control System (TACACS) protocol. When TACACS authentication is activated on the SSR, the user is prompted for a password when he or she tries to access Enable mode. The SSR queries a TACACS server to see if the password is valid. If the password is valid, the user is granted access to Enable mode.

Command Summary

Table 42 lists the **tacacs** commands. The sections following the table describe the command syntax.

Table 42. tacacs commands

tacacs enable
tacacs set host <*IPaddr*> **tacacs set [timeout** <*number*>**] [last-resort password** | **succeed]**
tacacs show stats | **all**

tacacs enable

Purpose

Enables TACACS authentication on the SSR. TACACS authentication is disabled by default on the SSR.

Format

tacacs enable

Mode

Configure

Description

The **tacacs enable** command starts TACACS authentication on the SSR. When you issue this command, the TACACS-related parameters set with **tacacs set** commands become active.

Parameters

None.

Restrictions

None.

Example

The following commands set TACACS-related parameters on the SSR. The commands are then activated with the **tacacs enable** command:

```
tacacs set host 207.135.89.15
tacacs set timeout 30
tacacs enable
```

tacacs set

Purpose

Sets parameters for authenticating the SSR through a TACACS server.

Format

tacacs set host <*IPaddr*>

tacacs set [timeout <*number*>**] [last-resort password | succeed]**

Mode

Configure

Description

The **tacacs set** command allows you to set TACACS-related parameters on the SSR, including the IP addresses of up to five TACACS servers, how long to wait for the TACACS server to authenticate the user, and what to do if the TACACS server does not reply by a given time.

Parameters

host <*IPaddr*> Is the IP address of a TACACS server. You can enter up to five TACACS servers. Enter one server per **tacacs set host** command.

timeout <*number*> Is the maximum time (in seconds) to wait for a TACACS server to reply. The default is 3 seconds.

last-resort Is the action to take if a TACACS server does not reply within the time specified by the **timeout** parameter. Specify one of the following:

- **password** The user is prompted for the Enable mode password set with **system set password** command (if one exists).
- **succeed** Access to the SSR is granted.

Restrictions

None.

Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are TACACS servers, and the SSR should wait no more than 30 seconds for a response from one of these servers. If a response from a TACACS server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the SSR **system set password** command.

```
tacacs set host 137.72.5.9
tacacs set host 137.72.5.41
tacacs set timeout 30
tacacs set last-resort password
```

tacacs show

Purpose

Displays information about TACACS configuration on the SSR.

Format

tacacs show stats | all

Mode

Enable

Description

The **tacacs show** command displays statistics and configuration parameters related to TACACS configuration on the SSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

Parameters

stats Displays the number of accepts, rejects, and timeouts for each TACACS server.

all Displays the configuration parameters set with the **tacacs set** command, in addition to the number of accepts, rejects, and timeouts for each TACACS server.

Restrictions

None.

Example

To display configuration parameters and TACACS server statistics:

```
tacacs show all
```

Chapter 58

tacacs-plus Commands

The **tacacs-plus** commands let you secure access to the SSR using the TACACS Plus protocol. When a user logs in to the SSR or tries to access Enable mode, he or she is prompted for a password. If TACACS Plus authentication is enabled on the SSR, it will contact a TACACS Plus server to verify the user. If the user is verified, he or she is granted access to the SSR.

Note: The SSR currently supports the Password Authentication Protocol (PAP) method of authentication but not the Challenge Handshake Authentication Protocol (CHAP) method.

Command Summary

Table 43 lists the **tacacs-plus** commands. The sections following the table describe the command syntax.

Table 43. tacacs-plus commands

tacacs-plus accounting command level <*level*>
tacacs-plus accounting shell start | stop | all
tacacs-plus accounting snmp active | startup
tacacs-plus accounting system fatal | error | warning | info
tacacs-plus authentication login | enable

Table 43. tacacs-plus commands (Continued)

tacacs-plus enable
tacacs-plus set server <*IPaddr*> **tacacs-plus set [timeout** <*number*>**] [key** <*string*>**] [last-resort password | succeed]**
tacacs-plus show stats | all

tacacs-plus accounting command level

Purpose

Causes the specified types of commands to be logged to the TACACS Plus server.

Format

tacacs-plus accounting command level <*level*>

Mode

Configure

Description

The **tacacs-plus accounting command level** command allows you specify the types of commands that are logged to the TACACS Plus server. The user ID and timestamp are also logged.

Parameters

<*level*> Specifies the type(s) of commands that are logged to the TACACS Plus server. Enter one of the following values:

- 5 Log Configure commands.
- 10 Log all Configure and Enable commands.
- 15 Log all Configure, Enable, and User commands.

Restrictions

None.

Example

To cause Configure, Enable, and User mode commands to be logged on the TACACS Plus server:

```
ssr(config)# tacacs-plus accounting command level 15
```

tacacs-plus accounting shell

Purpose

Causes an entry to be logged on the TACACS Plus server when a shell is stopped or started on the SSR.

Format

tacacs-plus accounting shell start | stop | all

Mode

Configure

Description

The **tacacs-plus accounting shell** command allows you to track shell usage on the SSR. It causes an entry to be logged on the TACACS Plus server when a shell is started or stopped. You can specify that an entry be logged when a shell is started, when a shell is stopped, or when a shell is either started or stopped.

Parameters

start Logs an entry when a shell is started.

stop Logs an entry when a shell is stopped

all Logs an entry when a shell is either started or stopped

Restrictions

None.

Example

To cause an entry to be logged on the TACACS Plus server when a shell is either started or stopped on the SSR:

```
ssr(config)# tacacs-plus accounting shell all
```

tacacs-plus accounting snmp

Purpose

Logs to the TACACS Plus server any changes made to the startup or active configuration via SNMP.

Format

tacacs-plus accounting snmp active | startup

Mode

Configure

Description

The **tacacs-plus accounting snmp** command allows you to track changes made to the active or startup configuration through SNMP. It causes an entry to be logged on the TACACS Plus server whenever a change is made to the ACL configuration. You can specify that an entry be logged to the active or startup configuration.

Parameters

active Logs an entry when a change is made to the active configuration.

startup Logs an entry when a change is made to the startup configuration.

Restrictions

None.

Example

To cause an entry to be logged on the TACACS Plus server whenever an ACL configuration change is made via SNMP to the active configuration:

```
ssr(config)# tacacs-plus accounting snmp active
```

tacacs-plus accounting system

Purpose

Specifies the type(s) of messages to be logged on the TACACS Plus server.

Format

tacacs-plus accounting system fatal | error | warning | info

Mode

Configure

Description

The **tacacs-plus accounting system** command allows you to specify the types of messages that are logged on the TACACS Plus server.

Parameters

fatal
Logs only fatal messages.

error
Logs fatal messages and error messages.

warning
Logs fatal messages, error messages, and warning messages.

info
Logs all messages, including informational messages.

Restrictions

None.

Example

To log only fatal and error messages on the TACACS Plus server:

```
ssr(config)# tacacs-plus accounting system error
```

tacacs-plus authentication

Purpose

Causes TACACS Plus authentication to be performed at either the SSR login prompt or when the user tries to access Enable mode.

Format

tacacs-plus authentication login | enable

Mode

Configure

Description

The **tacacs-plus authentication** command allows you to specify when TACACS Plus authentication is performed: either when a user logs in to the SSR, or tries to access Enable mode.

Parameters

login Authenticates users at the SSR login prompt.

enable Authenticates users when they try to access Enable mode.

Restrictions

None.

Example

To perform TACACS Plus authentication at the SSR login prompt:

```
ssr(config)# tacacs-plus authentication login
```

tacacs-plus enable

Purpose

Enables TACACS Plus authentication on the SSR. TACACS Plus authentication is disabled by default on the SSR.

Format

tacacs-plus enable

Mode

Configure

Description

The **tacacs-plus enable** command causes TACACS Plus authentication to be activated on the SSR. You set TACACS Plus-related parameters with the **tacacs-plus set**, **tacacs-plus accounting shell**, and **tacacs-plus authorization** commands, then use the **tacacs-plus enable** command to activate TACACS Plus authentication.

Parameters

None.

Restrictions

None.

Example

The following commands set TACACS Plus-related parameters on the SSR. The commands are then activated with the **tacacs-plus enable** command:

```
ssr(config)# tacacs-plus set server 207.135.89.15
ssr(config)# tacacs-plus set timeout 30
ssr(config)# tacacs-plus authentication login
ssr(config)# tacacs-plus accounting shell all
ssr(config)# tacacs-plus enable
```

tacacs-plus set

Purpose

Sets parameters for authenticating the SSR through a TACACS Plus server.

Format

tacacs-plus set server <*IPaddr*>

tacacs-plus set [timeout <*number*>**] [key** <*string*>**] [last-resort password | succeed]**

Mode

Configure

Description

The **tacacs-plus set** command allows you to set TACACS Plus-related parameters on the SSR, including the IP address of the TACACS Plus server, how long to wait for the TACACS Plus server to authenticate the user, an encryption key, and what to do if the TACACS Plus server does not reply by a given time.

Parameters

host <*IPaddr*> Is the IP address of a TACACS Plus server. You can enter up to five TACACS Plus servers. Enter one server per **tacacs-plus set server** command.

timeout <*number*> Is the maximum time (in seconds) to wait for a TACACS Plus server to reply. The default is 3 seconds.

key <*string*> Is an encryption key to be shared with the TACACS Plus server.

last-resort Is the action to take if a TACACS Plus server does not reply within the time specified by the **timeout** parameter. Specify one of the following:

- **password** The user is prompted for the password set with **system set password** command (if one has been set).
- **succeed** Access to the SSR is granted.

Restrictions

None.

Example

The following commands specify that hosts 137.72.5.9 and 137.72.5.41 are TACACS Plus servers, and the SSR should wait no more than 30 seconds for a response from one of these servers. If a response from a TACACS Plus server doesn't arrive in 30 seconds, the user is prompted for the password that was set with the SSR **system set password** command.

```
ssr(config)# tacacs-plus set server 137.72.5.9
ssr(config)# tacacs-plus set server 137.72.5.41
ssr(config)# tacacs-plus set timeout 30
ssr(config)# tacacs-plus set last-resort password
```

tacacs-plus show

Purpose

Displays information about TACACS Plus configuration on the SSR.

Format

tacacs-plus show stats | all

Mode

Enable

Description

The **tacacs-plus show** command displays statistics and configuration parameters related to TACACS Plus configuration on the SSR. The statistics displayed include:

accepts Number of times each server responded and validated the user successfully.

rejects Number of times each server responded and denied the user access, either because the user wasn't known, or the wrong password was supplied.

timeouts Number of times each server did not respond.

Parameters

stats Displays the accepts, rejects, and timeouts for each TACACS Plus server.

all Displays the configuration parameters set with the **tacacs-plus set** command, in addition to the accepts, rejects, and timeouts for each TACACS Plus server.

Restrictions

None.

Example

To display configuration parameters and TACACS Plus server statistics:

```
ssr# tacacs-plus show all
```

Chapter 59

telnet Command

The **telnet** command opens a Telnet session to the specified host.

Format

telnet <*hostname-or-IPaddr*> **[socket** <*socket-number*>**]**

Mode

User or Enable

Description

The **telnet** command allows you to open a Telnet session to the specified host.

Parameters

<*hostname-or-IPaddr*>
The host name or IP address of the remote computer that you want to access.

socket <*socket-number*>
The TCP port through which the Telnet session will be opened. If this parameter is not specified, the Telnet port (socket number 23) is assumed. This parameter can be used to test other ports; for example, socket number 21 is the port for FTP.

Restrictions

None.

Example

To open a Telnet session on the host “ssr4”:

```
ssr# telnet ssr4
```

Chapter 60

traceroute Command

The **traceroute** command traces the path a packet takes to reach a remote host.

Format

traceroute <*host*> **[max-ttl** <*num*>**] [probes** <*num*>**] [size** <*num*>**] [source** <*host*>**] [tos** <*num*>**] [wait-time** <*secs*>**] [verbose] [noroute]**

Mode

User

Description

The **traceroute** command traces the route taken by a packet to reach a remote IP host. The **traceroute** command examines the route taken by a packet traveling from a source to a destination. By default, the source of the packet is the SSR. However, one can specify a different source and track the route between it and a destination. The route is calculated by initially sending a probe (packet) from the source to the destination with a TTL of 1. Each intermediate router that is not able to reach the final destination directly will send back an ICMP Time Exceeded message. Subsequent probes from the source will increase the TTL value by 1. As each Time Exceeded message is received, the program keeps track of the address of each intermediate gateway. The probing stops when the packet reaches the destination or the TTL exceeds the **max-ttl** value.

Parameters

<host>
Hostname or IP address of the destination

max-ttl *<num>*
Maximum number of gateways (“hops”) to trace

probes *<num>*
Number of probes to send

size *<num>*
Packet size of each probe

source *<host>*
Hostname or IP address of the source

tos *<num>*
Type of Service value in the probe packet

wait-time *<secs>*
Maximum time to wait for a response

verbose
Displays results in verbose mode

noroute
Ignores the routing table and sends a probe to a host on a directly attached network. If the destination is not on the local network, an error is returned.

Restrictions

None.

Example

To display the route from the SSR to the host *othello* in verbose mode:

```
ssr# traceroute othello verbose
```

Chapter 61

vlan Commands

The vlan commands let you perform the following tasks:

- Create VLANs
- List VLANs
- Add ports to VLANs
- Change the port membership of VLANs
- Make a VLAN port either a trunk port or an access port

Command Summary

Table 44 lists the vlan commands. The sections following the table describe the command syntax.

Table 44. vlan commands

vlan add ports <*port-list*> **to** <*vlan-name*>
vlan create <*vlan-name*> <*type*> *id* <*num*>
vlan make <*port-type*> <*port-list*>
vlan show

vlan add ports

Purpose

Adds ports to a VLAN.

Format

vlan add ports <*port-list*> **to** <*vlan-name*>

Mode

Configure

Description

The **vlan add ports** command adds ports to an existing VLAN. You do not need to specify the VLAN type when you add ports. You specify the VLAN type when you create the VLAN (using the **vlan create** command).

Parameters

<*port-list*>
The ports you are adding to the VLAN. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

<*vlan-name*>
Name of the VLAN to which you are adding ports.

Restrictions

The VLAN to which you add ports must already exist. To create a VLAN, use the **vlan create** command. An access port can be added to only one IP VLAN, one IPX VLAN, and one bridged-protocols VLAN.

vlan create

Purpose

Creates a VLAN based on ports or protocol.

Format

vlan create <*vlan-name*> <*type*> **id <*num*>**

Mode

Configure

Description

The **vlan create** command creates a VLAN definition. You can create a port-based VLAN or a protocol-based VLAN.

Parameters

<*vlan-name*> Name of the VLAN. The VLAN name is a string up to 32 characters long.

Note: The VLAN name cannot begin with an underscore (_) or the word "SYS_". The names "control", "default", "blackhole", "reserved", and "learning" cannot be used.

<*type*> The type of VLAN you are adding. The VLAN type determines the types of traffic the SSR will forward on the VLAN. Specify any combination of the first seven types that follow ***or*** specify **port-based**:

ip
Create this VLAN for IP traffic

ipx
Create this VLAN for IPX traffic

appletalk
Create this VLAN for AppleTalk traffic

dec
Create this VLAN for DECnet traffic

sna
Create this VLAN for SNA traffic

ipv6
Create this VLAN for IPv6 traffic

bridged-protocols
Create this VLAN for extended VLAN types (DEC, SNA, Appletalk, IPv6), and non-IP and non-IPX protocols

Note: You can specify a combination of **ip**, **ipx**, **appletalk**, **dec**, **sna**, **ipv6**, and **bridged-protocols**. If you specify *any* of the extended VLAN types (**sna**, **dec**, **appletalk**, **ipv6**) with the **bridged-protocols** option, then all the other extended VLAN types are removed from the VLAN. See the following table:

Configuration Command	Protocols Included in VLAN	Protocols Excluded from VLAN
vlan create <*vlan-name*> ip	**IP**	**IPX, SNA, IPv6, DECnet, Appletalk, Other**
vlan create <*vlan-name*> ip bridged-protocols	**IP, SNA, DECnet, IPv6, Appletalk, Other**	**IPX**
vlan create <*vlan-name*> ip bridged-protocols sna	**IP, SNA, Other**	**IPX, IPv6, DECnet, Appletalk**
vlan create <*vlan-name*> ip bridged-protocols sna ipv6	**IP, SNA, IPv6, Other**	**IPX, DECnet, Appletalk**

port-based
Create this VLAN for all the traffic types listed above (port-based VLAN)

Note: You can specify a combination of **ip**, **ipx**, **appletalk**, **dec**, **sna**, **ipv6**, and **bridged-protocols** *or* you can specify **port-based**; you cannot specify **port-based** with any of the other options.

id <*num*> ID of this VLAN. The ID must be unique. You can specify a number from 2 – 4093. If more than one SSR will be configured with the same VLAN, you must specify the same VLAN ID on each SSR.

Restrictions

The following *cannot* be used for VLAN names:

- control

- default
- blackhole
- reserved
- learning
- names starting with an underscore (_) or “sys_”

Examples

The following command creates a VLAN ‘blue’ for IP, SNA, non-IPX, non-DECnet, non-Appletalk, non-IPv6 protocols.:

```
ssr(config)# vlan create blue ip bridged-protocols sna
```

The following command creates a VLAN ‘red’ for IP, non-IPX, and extended VLAN types SNA, DECnet, Appletalk, and IPv6:

```
ssr(config)# vlan create red ip bridged-protocols
```

vlan make

Purpose

Configures the specified ports into either trunk or access ports.

Format

vlan make <*port-type*> <*port-list*>

Mode

Configure

Description

The **vlan make** command turns a port into a VLAN trunk or VLAN access port. A VLAN trunk port can forward traffic for multiple VLANs. Use trunk ports when you want to connect SSR switches together and send traffic for multiple VLANs on a single network segment connecting the switches.

Parameters

<*port-type*>
The port type. You can specify one of the following types:

trunk-port
The port will forward traffic for multiple VLANs. The SSR will encapsulate all traffic in IEEE 802.1Q tag headers.

access-port
The port will forward traffic only for the VLANs to which you have added the ports and the traffic will be untagged. This is the default.

<*port-list*>
The ports you are configuring. You can specify a single port or a comma-separated list of ports. Example: et.1.3,et.(1-3).(4,6-8).

Restrictions

None.

vlan show

Purpose

Displays a list of all VLANs active on the SSR.

Format

vlan show

Mode

User or Enable

Description

The **vlan show** command lists all the VLANs that have been configured on the SSR.

Parameters

None.

Restrictions

None.

Chapter 62

web-cache Commands

The **web-cache** commands allow you to transparently redirect HTTP request to a group of local cache servers. This feature can provide faster user responses and reduce demands for WAN bandwidth.

Command Summary

Table 45 lists the **web-cache** commands. The sections following the table describe the command syntax.

Table 45. web-cache commands

web-cache <*cache-name*> **apply interface** <*interface-name*>
web-cache clear all | **cache-name** <*cache-name*>
web-cache <*cache-name*> **create bypass-list range** <*ipaddr-range*> | **list** <*ipaddr-list*> | **acl** <*acl-name*>
web-cache <*cache-name*> **create server-list** <*server-list-name*> **range** <*ipaddr-range*> | **list** <*ipaddr-list*>
web-cache <*cache-name*> **permit** | **deny hosts range** <*ipadd-range*> | **list** <*ipaddr-list*> | **acl** <*acl-name*>
web-cache <*cache-name*> **set http-port** <*port number*>
web-cache <*cache-name*> **set round-robin range** <*ipaddr-range*> | **list** <*ipaddr-list*>
web-cache show [all] [cache-name <*cache-name*> | **all] [servers cache** <*cache-name*> | **all]**

web-cache apply interface

Purpose

Applies a caching policy to an interface.

Format

web-cache <*cache-name*> **apply interface** <*interface-name*>

Mode

Configure

Description

The **web-cache apply** command lets you apply a configured cache policy to an outbound interface to start the redirection. The interface to which the cache policy is applied is typically the interface that connects to the Internet. This command redirects outbound HTTP traffic to the cache servers.

Parameters

<*cache-name*>
: The name of a cache policy configured with the **web-cache create server-list** command.

<*interface-name*>
: The name of the outbound interface that connects to the actual Web server. Typically, this is the interface that connects to the Internet.

Restrictions

None.

Example

To apply the caching policy ‘websrv1’ to the interface ‘inet2’:

```
ssr(config)# web-cache websrv1 apply interface inet2
```

web-cache clear

Purpose

Clears statistics for the specified caching policy.

Format

web-cache clear all | cache-name <*cache-name*>

Mode

Enable

Description

The **web-cache clear** command lets you clear statistics for all caching policies or for specified policies.

Parameters

all
Clears statistics for all caching policies.

cache-name <*cache-name*>

Clears statistics for the specified caching policy.

Restrictions

None.

Examples

To clear statistics for the caching policy 'websrv1':

```
ssr# web-cache clear cache-name websrv1
```

web-cache create bypass-list

Purpose

Defines the destination sites for which HTTP requests are not redirected to the cache servers, but sent direct.

Format

web-cache <*cache-name*> **create bypass-list range** <*ipaddr-range*> | **list** <*ipaddr-list*> | **acl** <*acl-name*>

Mode

Configure

Description

Certain web sites require authentication of source IP addresses for user access. Requests to these sites cannot be sent to the cache servers. The **web-cache create bypass-list** command allows you to define the destinations to which HTTP requests must be sent directly without redirection to a cache server. You can specify a range of IP addresses, a list of up to four IP addresses, or an ACL that qualifies these hosts.

Parameters

<*cache-name*>
The name of the caching policy for which the specified hosts will not apply.

range <*ipaddr-range*>
A range of host IP addresses in the form "176.89.10.10 176.89.10.50". This adds the hosts 176.89.10.10, 176.89.10.11, etc., through 176.89.10.50 to the bypass list.

list <*ipaddr-list*>
A list of up to four destination IP addresses in the form "176.89.10.10 176.89.10.11 176.89.10.12".

acl <*acl-name*>
Name of the ACL profile that defines the packet profile to bypass. The ACL may contain either **permit** or **deny** keywords. The **web-cache create bypass-list** command only looks at the following ACL rule parameter values: protocol, source IP address, destination IP address, source port, destination port, and TOS.

Restrictions

None.

Examples

To specify the hosts 176.89.10.10 and 176.89.10.11 for the bypass list for the caching policy ‘websrv1’:

```
ssr(config)# web-cache websrv1 create bypass-list list “176.89.10.10
176.89.10.11”
```

To specify the hosts defined in the ACL ‘nocache’ for the bypass list for the caching policy ‘websrv1’:

```
ssr(config)# web-cache websrv1 create bypass-list acl nocache
```

web-cache create server-list

Purpose

Defines the list of servers to be used for caching.

Format

web-cache <*cache-name*> **create server-list** <*server-list-name*> **range** <*ipaddr-range*> | **list** <*ipaddr-list*>

Mode

Configure

Description

The **web-cache create server-list** command allows you to create a group of servers that are used for the specified caching policy. If there are multiple cache servers, load balancing is done based on the destination IP address. If any cache server fails, traffic is redirected to other active servers. You can specify either a range of IP addresses or a list of up to four IP addresses. Note that traffic that is sent from a server in the server list is not redirected.

Parameters

<*cache-name*>
The name of the caching policy.

<*server-list-name*>
The name of this list of servers.

range <*ipaddr-range*>
A range of host IP addresses in the form "176.89.10.10 176.89.10.50". This adds the hosts 176.89.10.10, 176.89.10.11, etc., through 176.89.10.50 to the server list.

list <*ipaddr-list*>
A list of up to four host IP addresses in the form "176.89.10.10 176.89.10.11 176.89.10.12".

Restrictions

None.

Examples

To specify the server list ‘servers1’ for the caching policy ‘websrv1’:

```
ssr(config)# web-cache websrv1 create server-list servers1 range
“10.10.10.10 10.10.10.50”
```

web-cache permit|deny hosts

Purpose

Specifies the hosts whose HTTP requests are redirected to the cache servers.

Format

web-cache <*cache-name*> **permit | deny hosts range** <*ipadd-range*> | **list** <*ipaddr-list*> | **acl** <*acl-name*>

Mode

Configure

Description

The **web-cache permit** command lets you specify the hosts (users) whose HTTP requests are redirected to the cache servers, while the **web-cache deny** command lets you specify the hosts whose HTTP requests are not redirected to the cache servers. If no **permit** command is specified, all HTTP requests are redirected to the cache servers. You can specify a range of IP addresses, a list of up to four IP addresses, or an ACL that qualifies these hosts.

Parameters

<*cache-name*>
The name of the cache.

range <*ipaddr-range*>
A range of host IP addresses in the form "176.89.10.10 176.89.10.50".

list <*ipaddr-list*>
A list of up to four host IP addresses in the form "176.89.10.10 176.89.10.11 176.89.10.12".

acl <*acl-name*>
Name of the ACL profile to be used. This defines the profile of the packets to be permitted or denied. The **web-cache permit/deny** command only looks at the following ACL rule parameter values: protocol, source IP address, destination IP address, source port, destination port, and TOS.

Restrictions

None.

Examples

To allow the HTTP requests of certain hosts to be redirected to the cache servers:

```
ssr(config)# web-cache websrv1 permit hosts range “10.10.20.10 10.10.20.50”
```

To specify that the HTTP requests of certain hosts not be redirected to the cache servers:

```
ssr(config)# web-cache websrv1 deny hosts list “10.10.20.61 10.10.20.75”
```

web-cache set http-port

Purpose

Specifies the HTTP port used by a proxy server.

Format

web-cache <*cache-name*> **set http-port** <*port number*>

Mode

Configure

Description

Some networks use proxy servers that listen for HTTP requests on a non-standard port number. The SSR can be configured to redirect HTTP requests on a non-standard HTTP port. The **web-cache set http-port** command lets you specify the port number that is used by the proxy server for HTTP requests. The default is port 80.

Parameters

<*cache-name*>
The name of the cache.

<*port number*>
Specifies the port number used by the proxy server for HTTP requests. Specify a value between 1 and 65535.

Restrictions

None.

Example

To set the port number for HTTP requests:

```
ssr(config)# web-cache websvr1 set http-port 100
```

web-cache set round-robin

Purpose

Specifies a list of destination IP addresses to be distributed across cache servers.

Format

web-cache <*cache-name*> **set round-robin range** <*ipaddr-range*> | **list** <*ipaddr-list*>

Mode

Configure

Description

The SSR determines the cache server to redirect an HTTP request, based on the destination IP address of the request. If a certain web site is accessed very frequently, the cache server that services HTTP requests to this web site can become overloaded with user requests. The **web-cache set round-robin** command allows you to distribute destination IP addresses for HTTP requests across cache servers in a round-robin manner. If a cache server fails, the address range associated with that server is redistributed among the remaining servers.

Parameters

<*cache-name*>
The name of the caching policy.

range <*ipaddr-range*>
A range of host IP addresses in the form "176.89.10.10 176.89.10.50".

list <*ipaddr-list*>
A list of up to four destination IP addresses in the form "176.89.10.10 176.89.10.11 176.89.10.12".

Restrictions

None.

Example

To specify destination IP addresses to be distributed across the caching policy ‘websvr1’ servers:

```
ssr(config)# web-cache set round-robin list “176.20.20.10 176.20.50.60”
```

web-cache show

Purpose

Displays information about caching policies.

Format

web-cache show [all] [cache-name <*cache-name*> **| all] [servers cache** <*cache-name*> **| all]**

Mode

Enable

Description

The **web-cache show** command allows you to display web caching information for specific caching policies or server lists.

Parameters

all
Displays all web cache information for all caching policies and all server lists.

cache-name <*cache-name*> **| all**
Displays web cache information for the specified caching policy. **all** displays all caching policies.

servers cache <*cache-name*> **| all**
Displays information for the servers configured for the specified caching policy. **all** displays all configured cache servers.

Restrictions

None.

Examples

To display web cache information for a specific caching policy:

```
ssr# web-cache show cache-name cache1
Cache Name : cache1 ❶
Applied Interfaces : ip1 ❷
Bypass list : none ❸
HTTP Port : 80 ❹

 ❺             ❻                ❼             ❽          ❾    ❿  ⓫
ACL      Source IP/Mask   Dest. IP/Mask   SrcPort   DstPort   TOS Port
---      --------------   -------------   --------- --------- --- ----
deny207  172.89.1.1/32    207.135.0.0/16  any       http      0   IP

   ⓬       ⓭          ⓮
Server   Max con IP address
---- --  ------- ----------
s1       2000    176.89.10.50 - 176.89.10.60

Access Users ⓯
------ -----
Permit All Users
Deny   profile deny207
```

Legend:

1. The name of the cache policy.
2. The outbound interface where the cache policy was applied, typically an interface that connects to the Internet.
3. Destination sites for which HTTP requests are *not* redirected to cache servers and are sent direct.
4. The HTTP port used by a proxy server. A port number other than 80 can be specified with the **web-cache set http-port** command.
5. The names of the profiles (created with an **acl** statement) associated with this cache policy.
6. The source address and filtering mask.
7. The destination address and filtering mask.
8. The source port.
9. The destination port.
10. The TOS value in the packet.
11. The protocol.
12. The server list name.

13. The maximum number of connections that can be handled by each server in the server list.
14. The list or range of IP addresses of the servers in the server list.
15. The hosts (users) whose HTTP requests *are* redirected to the cache servers and the hosts whose HTTP requests are *not* redirected to the cache servers. If no **permit** command is specified, all HTTP requests are redirected to the cache servers.

To display information for all configured web cache servers:

```
ssr# web-cache show servers cache cache1
Cache name : cache1 ❶

  ❷        ❸           ❹         ❺         ❻
Block IP address     Max Conn  Used Cnt  Status
----- ----------     --------  --------  ------
s1    176.89.10.50   2000      0         Down
s1    176.89.10.51   2000      0         Down
s1    176.89.10.52   2000      0         Down
s1    176.89.10.53   2000      0         Down
s1    176.89.10.54   2000      0         Down
s1    176.89.10.55   2000      0         Down
s1    176.89.10.56   2000      0         Down
s1    176.89.10.57   2000      0         Down
s1    176.89.10.58   2000      0         Down
s1    176.89.10.59   2000      0         Down
s1    176.89.10.60   2000      0         Down
```

Legend:

1. The name of the cache policy.
2. The server list name.
3. The IP address of a server in the server list.
4. The maximum number of connections that can be handled by the server.
5. The number of connections currently being handled by the server.
6. The current status of the server.

Appendix A

RMON 2 Protocol Directory

This appendix lists the protocol encapsulations that can be managed with the RMON 2 Protocol Directory group on the SSR. You can specify protocol encapsulations with the **rmon set protocol-directory** or **rmon show protocol-directory** commands. For example, ether2.ipx specifies IPX over Ethernet II, while *ether2.ipx specifies IPX over any link layer protocol. The protocol object IDs are defined in RFC 2074.

The protocols are listed in the following order:

- Ethernet Applications
- IP (version 4) Applications
- IPX Applications
- TCP Applications
- UDP Applications

Protocol Encapsulation	Protocol Identifier (Object ID)
Ethernet Applications	
ether2.idp	8.0.0.0.1.0.0.6.0.2.0.0
ether2.ip-v4	8.0.0.0.1.0.0.8.0.2.0.0
ether2.chaosnet	8.0.0.0.1.0.0.8.4.2.0.0
ether2.arp	8.0.0.0.1.0.0.8.6.2.0.0
ether2.vip	8.0.0.0.1.0.0.11.173.2.0.0
ether2.vloop	8.0.0.0.1.0.0.11.174.2.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
ether2.vecho	8.0.0.0.1.0.0.11.175.2.0.0
ether2.netbios-3com	8.0.0.0.1.0.0.60.0.2.0.0
ether2.dec	8.0.0.0.1.0.0.96.0.2.0.0
ether2.mop	8.0.0.0.1.0.0.96.1.2.0.0
ether2.mop2	8.0.0.0.1.0.0.96.2.2.0.0
ether2.drp	8.0.0.0.1.0.0.96.3.2.0.0
ether2.lat	8.0.0.0.1.0.0.96.4.2.0.0
ether2.dec-diag	8.0.0.0.1.0.0.96.5.2.0.0
ether2.lavc	8.0.0.0.1.0.0.96.7.2.0.0
ether2.rarp	8.0.0.0.1.0.0.128.53.2.0.0
ether2.atalk	8.0.0.0.1.0.0.128.155.2.0.0
ether2.vloop2	8.0.0.0.1.0.0.128.196.2.0.0
ether2.vecho2	8.0.0.0.1.0.0.128.197.2.0.0
ether2.sna-th	8.0.0.0.1.0.0.128.213.2.0.0
ether2.aarp	8.0.0.0.1.0.0.128.243.2.0.0
ether2.ipx	8.0.0.0.1.0.0.129.55.2.0.0
ether2.snmp	8.0.0.0.1.0.0.129.76.2.0.0
ether2.ip-v6	8.0.0.0.1.0.0.134.221.2.0.0
ether2.loopback	8.0.0.0.1.0.0.144.0.2.0.0
*ether2.ip-v4	8.1.0.0.1.0.0.8.0.2.0.1
*ether2.ipx	8.1.0.0.1.0.0.129.55.2.0.0
IP (version 4) Applications	
*ether2.ip-v4.icmp	12.1.0.0.1.0.0.8.0.0.0.0.1.3.0.1.0
*ether2.ip-v4.igmp	12.1.0.0.1.0.0.8.0.0.0.0.2.3.0.1.0
*ether2.ip-v4.ggp	12.1.0.0.1.0.0.8.0.0.0.0.3.3.0.1.0
*ether2.ip-v4.ipip4	12.1.0.0.1.0.0.8.0.0.0.0.4.3.0.1.0
*ether2.ip-v4.st	12.1.0.0.1.0.0.8.0.0.0.0.5.3.0.1.0
*ether2.ip-v4.tcp	12.1.0.0.1.0.0.8.0.0.0.0.6.3.0.1.0
*ether2.ip-v4.ucl	12.1.0.0.1.0.0.8.0.0.0.0.7.3.0.1.0
*ether2.ip-v4.egp	12.1.0.0.1.0.0.8.0.0.0.0.8.3.0.1.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.igp	12.1.0.0.1.0.0.8.0.0.0.0.9.3.0.1.0
*ether2.ip-v4.bbn-rcc-mon	12.1.0.0.1.0.0.8.0.0.0.0.10.3.0.1.0
*ether2.ip-v4.nvp2	12.1.0.0.1.0.0.8.0.0.0.0.11.3.0.1.0
*ether2.ip-v4.pup	12.1.0.0.1.0.0.8.0.0.0.0.12.3.0.1.0
*ether2.ip-v4.argus	12.1.0.0.1.0.0.8.0.0.0.0.13.3.0.1.0
*ether2.ip-v4.emcon	12.1.0.0.1.0.0.8.0.0.0.0.14.3.0.1.0
*ether2.ip-v4.xnet	12.1.0.0.1.0.0.8.0.0.0.0.15.3.0.1.0
*ether2.ip-v4.chaos	12.1.0.0.1.0.0.8.0.0.0.0.16.3.0.1.0
*ether2.ip-v4.udp	12.1.0.0.1.0.0.8.0.0.0.0.17.3.0.1.0
*ether2.ip-v4.mux	12.1.0.0.1.0.0.8.0.0.0.0.18.3.0.1.0
*ether2.ip-v4.dcn-meas	12.1.0.0.1.0.0.8.0.0.0.0.19.3.0.1.0
*ether2.ip-v4.hmp	12.1.0.0.1.0.0.8.0.0.0.0.20.3.0.1.0
*ether2.ip-v4.prm	12.1.0.0.1.0.0.8.0.0.0.0.21.3.0.1.0
*ether2.ip-v4.xns-idp	12.1.0.0.1.0.0.8.0.0.0.0.22.3.0.1.0
*ether2.ip-v4.trunk-1	12.1.0.0.1.0.0.8.0.0.0.0.23.3.0.1.0
*ether2.ip-v4.trunk-2	12.1.0.0.1.0.0.8.0.0.0.0.24.3.0.1.0
*ether2.ip-v4.leaf-1	12.1.0.0.1.0.0.8.0.0.0.0.25.3.0.1.0
*ether2.ip-v4.leaf-2	12.1.0.0.1.0.0.8.0.0.0.0.26.3.0.1.0
*ether2.ip-v4.rdp	12.1.0.0.1.0.0.8.0.0.0.0.27.3.0.1.0
*ether2.ip-v4.irtp	12.1.0.0.1.0.0.8.0.0.0.0.28.3.0.1.0
*ether2.ip-v4.iso-tp4	12.1.0.0.1.0.0.8.0.0.0.0.29.3.0.1.0
*ether2.ip-v4.netbit	12.1.0.0.1.0.0.8.0.0.0.0.30.3.0.1.0
*ether2.ip-v4.mfe-nsp	12.1.0.0.1.0.0.8.0.0.0.0.31.3.0.1.0
*ether2.ip-v4.merit-inp	12.1.0.0.1.0.0.8.0.0.0.0.32.3.0.1.0
*ether2.ip-v4.sep	12.1.0.0.1.0.0.8.0.0.0.0.33.3.0.1.0
*ether2.ip-v4.third-pc	12.1.0.0.1.0.0.8.0.0.0.0.34.3.0.1.0
*ether2.ip-v4.idpr	12.1.0.0.1.0.0.8.0.0.0.0.35.3.0.1.0
*ether2.ip-v4.xtp	12.1.0.0.1.0.0.8.0.0.0.0.36.3.0.1.0
*ether2.ip-v4.ddp	12.1.0.0.1.0.0.8.0.0.0.0.37.3.0.1.0
*ether2.ip-v4.idpr-cmtp	12.1.0.0.1.0.0.8.0.0.0.0.38.3.0.1.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tp-plus-plus	12.1.0.0.1.0.0.8.0.0.0.0.39.3.0.1.0
*ether2.ip-v4.il	12.1.0.0.1.0.0.8.0.0.0.0.40.3.0.1.0
*ether2.ip-v4.sip	12.1.0.0.1.0.0.8.0.0.0.0.41.3.0.1.0
*ether2.ip-v4.sdrp	12.1.0.0.1.0.0.8.0.0.0.0.42.3.0.1.0
*ether2.ip-v4.sip-sr	12.1.0.0.1.0.0.8.0.0.0.0.43.3.0.1.0
*ether2.ip-v4.sip-frag	12.1.0.0.1.0.0.8.0.0.0.0.44.3.0.1.0
*ether2.ip-v4.idrp	12.1.0.0.1.0.0.8.0.0.0.0.45.3.0.1.0
*ether2.ip-v4.rsvp	12.1.0.0.1.0.0.8.0.0.0.0.46.3.0.1.0
*ether2.ip-v4.gre	12.1.0.0.1.0.0.8.0.0.0.0.47.3.0.1.0
*ether2.ip-v4.mhrp	12.1.0.0.1.0.0.8.0.0.0.0.48.3.0.1.0
*ether2.ip-v4.bna	12.1.0.0.1.0.0.8.0.0.0.0.49.3.0.1.0
*ether2.ip-v4.sipp-esp	12.1.0.0.1.0.0.8.0.0.0.0.50.3.0.1.0
*ether2.ip-v4.sipp-ah	12.1.0.0.1.0.0.8.0.0.0.0.51.3.0.1.0
*ether2.ip-v4.i-nlsp	12.1.0.0.1.0.0.8.0.0.0.0.52.3.0.1.0
*ether2.ip-v4.swipe	12.1.0.0.1.0.0.8.0.0.0.0.53.3.0.1.0
*ether2.ip-v4.nhrp	12.1.0.0.1.0.0.8.0.0.0.0.54.3.0.1.0
*ether2.ip-v4.priv-host	12.1.0.0.1.0.0.8.0.0.0.0.61.3.0.1.0
*ether2.ip-v4.cftp	12.1.0.0.1.0.0.8.0.0.0.0.62.3.0.1.0
*ether2.ip-v4.priv-net	12.1.0.0.1.0.0.8.0.0.0.0.63.3.0.1.0
*ether2.ip-v4.sat-expak	12.1.0.0.1.0.0.8.0.0.0.0.64.3.0.1.0
*ether2.ip-v4.kryptolan	12.1.0.0.1.0.0.8.0.0.0.0.65.3.0.1.0
*ether2.ip-v4.rvd	12.1.0.0.1.0.0.8.0.0.0.0.66.3.0.1.0
*ether2.ip-v4.ippc	12.1.0.0.1.0.0.8.0.0.0.0.67.3.0.1.0
*ether2.ip-v4.priv-distfile	12.1.0.0.1.0.0.8.0.0.0.0.68.3.0.1.0
*ether2.ip-v4.sat-mon	12.1.0.0.1.0.0.8.0.0.0.0.69.3.0.1.0
*ether2.ip-v4.visa	12.1.0.0.1.0.0.8.0.0.0.0.70.3.0.1.0
*ether2.ip-v4.ipcv	12.1.0.0.1.0.0.8.0.0.0.0.71.3.0.1.0
*ether2.ip-v4.cpnx	12.1.0.0.1.0.0.8.0.0.0.0.72.3.0.1.0
*ether2.ip-v4.cphb	12.1.0.0.1.0.0.8.0.0.0.0.73.3.0.1.0
*ether2.ip-v4.wsn	12.1.0.0.1.0.0.8.0.0.0.0.74.3.0.1.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.pvp	12.1.0.0.1.0.0.8.0.0.0.0.75.3.0.1.0
*ether2.ip-v4.br-sat-mon	12.1.0.0.1.0.0.8.0.0.0.0.76.3.0.1.0
*ether2.ip-v4.sun-nd	12.1.0.0.1.0.0.8.0.0.0.0.77.3.0.1.0
*ether2.ip-v4.wb-mon	12.1.0.0.1.0.0.8.0.0.0.0.78.3.0.1.0
*ether2.ip-v4.wb-expak	12.1.0.0.1.0.0.8.0.0.0.0.79.3.0.1.0
*ether2.ip-v4.iso-ip	12.1.0.0.1.0.0.8.0.0.0.0.80.3.0.1.0
*ether2.ip-v4.vmtp	12.1.0.0.1.0.0.8.0.0.0.0.81.3.0.1.0
*ether2.ip-v4.secure-mvtp	12.1.0.0.1.0.0.8.0.0.0.0.82.3.0.1.0
*ether2.ip-v4.vines	12.1.0.0.1.0.0.8.0.0.0.0.83.3.0.1.0
*ether2.ip-v4.ttp	12.1.0.0.1.0.0.8.0.0.0.0.84.3.0.1.0
*ether2.ip-v4.nfsnet-igp	12.1.0.0.1.0.0.8.0.0.0.0.85.3.0.1.0
*ether2.ip-v4.dgp	12.1.0.0.1.0.0.8.0.0.0.0.86.3.0.1.0
*ether2.ip-v4.tcf	12.1.0.0.1.0.0.8.0.0.0.0.87.3.0.1.0
*ether2.ip-v4.igrp	12.1.0.0.1.0.0.8.0.0.0.0.88.3.0.1.0
*ether2.ip-v4.ospf	12.1.0.0.1.0.0.8.0.0.0.0.89.3.0.1.0
*ether2.ip-v4.sprite-rpc	12.1.0.0.1.0.0.8.0.0.0.0.90.3.0.1.0
*ether2.ip-v4.larp	12.1.0.0.1.0.0.8.0.0.0.0.91.3.0.1.0
*ether2.ip-v4.mtp	12.1.0.0.1.0.0.8.0.0.0.0.92.3.0.1.0
*ether2.ip-v4.ax-25	12.1.0.0.1.0.0.8.0.0.0.0.93.3.0.1.0
*ether2.ip-v4.ipip	12.1.0.0.1.0.0.8.0.0.0.0.94.3.0.1.0
*ether2.ip-v4.micp	12.1.0.0.1.0.0.8.0.0.0.0.95.3.0.1.0
*ether2.ip-v4.scc-sp	12.1.0.0.1.0.0.8.0.0.0.0.96.3.0.1.0
*ether2.ip-v4.etherip	12.1.0.0.1.0.0.8.0.0.0.0.97.3.0.1.0
*ether2.ip-v4.encap	12.1.0.0.1.0.0.8.0.0.0.0.98.3.0.1.0
*ether2.ip-v4.priv-encript	12.1.0.0.1.0.0.8.0.0.0.0.99.3.0.1.0
*ether2.ip-v4.gmtp	12.1.0.0.1.0.0.8.0.0.0.0.100.3.0.1.0
IPX Applications	
*ether2.ipx.nov-pep	12.1.0.0.1.0.0.129.55.0.0.0.0.3.0.0.0
*ether2.ipx.nov-pep.ncp	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.81.4.0.0.0.0
*ether2.ipx.nov-pep.nov-sap	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.82.4.0.0.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ipx.nov-pep.nov-rip	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.83.4.0.0.0.0
*ether2.ipx.nov-pep.nov-netbios	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.85.4.0.0.0.0
*ether2.ipx.nov-pep.nov-diag	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.86.4.0.0.0.0
*ether2.ipx.nov-pep.nov-sec	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.4.87.4.0.0.0.0
*ether2.ipx.nov-pep.smb	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.5.80.4.0.0.0.0
*ether2.ipx.nov-pep.smb2	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.5.82.4.0.0.0.0
*ether2.ipx.nov-pep.burst	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.13.5.4.0.0.0.0
*ether2.ipx.nov-pep.nov-watchdog	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.64.4.4.0.0.0.0
*ether2.ipx.nov-pep.nov-bcast	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.64.5.4.0.0.0.0
*ether2.ipx.nov-pep.nlsp	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.144.1.4.0.0.0.0
*ether2.ipx.nov-pep.snmp	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.144.15.4.0.0.0.0
*ether2.ipx.nov-pep.snmptrap	16.1.0.0.1.0.0.129.55.0.0.0.0.0.0.144.16.4.0.0.0.0
*ether2.ipx.nov-rip	12.1.0.0.1.0.0.129.55.0.0.0.1.3.0.0.0
*ether2.ipx.nov-echo	12.1.0.0.1.0.0.129.55.0.0.0.2.3.0.0.0
*ether2.ipx.nov-error	12.1.0.0.1.0.0.129.55.0.0.0.3.3.0.0.0
*ether2.ipx.nov-pep2	12.1.0.0.1.0.0.129.55.0.0.0.4.3.0.0.0
*ether2.ipx.nov-spx	12.1.0.0.1.0.0.129.55.0.0.0.5.3.0.0.0
*ether2.ipx.nov-pep3	12.1.0.0.1.0.0.129.55.0.0.0.17.3.0.0.0
*ether2.ipx.nov-netbios	12.1.0.0.1.0.0.129.55.0.0.0.20.3.0.0.0
TCP Applications	
*ether2.ip-v4.tcp.tcpmux	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.1.4.0.1.0.0
*ether2.ip-v4.tcp.compressnet-mgmt	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.2.4.0.1.0.0
*ether2.ip-v4.tcp.compressnet	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.3.4.0.1.0.0
*ether2.ip-v4.tcp.rje	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.5.4.0.1.0.0
*ether2.ip-v4.tcp.echo	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.7.4.0.1.0.0
*ether2.ip-v4.tcp.discard	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.9.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.systat	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.11.4.0.1.0.0
*ether2.ip-v4.tcp.daytime	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.13.4.0.1.0.0
*ether2.ip-v4.tcp.qotd	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.17.4.0.1.0.0
*ether2.ip-v4.tcp.msp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.18.4.0.1.0.0
*ether2.ip-v4.tcp.chargen	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.19.4.0.1.0.0
*ether2.ip-v4.tcp.ftp-data	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.20.4.0.1.0.0
*ether2.ip-v4.tcp.ftp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.21.4.0.1.0.0
*ether2.ip-v4.tcp.telnet	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.23.4.0.1.0.0
*ether2.ip-v4.tcp.priv-mail	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.24.4.0.1.0.0
*ether2.ip-v4.tcp.smtp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.25.4.0.1.0.0
*ether2.ip-v4.tcp.nsw-fe	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.27.4.0.1.0.0
*ether2.ip-v4.tcp.msg-icp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.29.4.0.1.0.0
*ether2.ip-v4.tcp.msg-auth	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.31.4.0.1.0.0
*ether2.ip-v4.tcp.dsp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.33.4.0.1.0.0
*ether2.ip-v4.tcp.priv-print	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.35.4.0.1.0.0
*ether2.ip-v4.tcp.time	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.37.4.0.1.0.0
*ether2.ip-v4.tcp.rap	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.38.4.0.1.0.0
*ether2.ip-v4.tcp.graphics	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.41.4.0.1.0.0
*ether2.ip-v4.tcp.nicname	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.43.4.0.1.0.0
*ether2.ip-v4.tcp.mpm-flags	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.44.4.0.1.0.0
*ether2.ip-v4.tcp.mpm	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.45.4.0.1.0.0
*ether2.ip-v4.tcp.mpm-send	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.46.4.0.1.0.0
*ether2.ip-v4.tcp.ni-ftp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.47.4.0.1.0.0
*ether2.ip-v4.tcp.auditd	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.48.4.0.1.0.0
*ether2.ip-v4.tcp.tacacs	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.49.4.0.1.0.0
*ether2.ip-v4.tcp.xns-time	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.52.4.0.1.0.0
*ether2.ip-v4.tcp.domain	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.53.4.0.1.0.0
*ether2.ip-v4.tcp.xns-ch	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.54.4.0.1.0.0
*ether2.ip-v4.tcp.isi-gl	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.55.4.0.1.0.0
*ether2.ip-v4.tcp.xns-auth	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.56.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.priv-term	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.57.4.0.1.0.0
*ether2.ip-v4.tcp.xns-mail	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.58.4.0.1.0.0
*ether2.ip-v4.tcp.priv-file	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.59.4.0.1.0.0
*ether2.ip-v4.tcp.ni-mail	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.61.4.0.1.0.0
*ether2.ip-v4.tcp.acas	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.62.4.0.1.0.0
*ether2.ip-v4.tcp.covia	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.64.4.0.1.0.0
*ether2.ip-v4.tcp.tacacs-ds	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.65.4.0.1.0.0
*ether2.ip-v4.tcp.sql*net	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.66.4.0.1.0.0
*ether2.ip-v4.tcp.gopher	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.70.4.0.1.0.0
*ether2.ip-v4.tcp.netrjs-1	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.71.4.0.1.0.0
*ether2.ip-v4.tcp.netrjs-2	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.72.4.0.1.0.0
*ether2.ip-v4.tcp.netrjs-3	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.73.4.0.1.0.0
*ether2.ip-v4.tcp.netrjs-4	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.74.4.0.1.0.0
*ether2.ip-v4.tcp.priv-dialout	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.75.4.0.1.0.0
*ether2.ip-v4.tcp.deos	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.76.4.0.1.0.0
*ether2.ip-v4.tcp.priv-rje	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.77.4.0.1.0.0
*ether2.ip-v4.tcp.vettcp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.78.4.0.1.0.0
*ether2.ip-v4.tcp.finger	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.79.4.0.1.0.0
*ether2.ip-v4.tcp.www-http	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.80.4.0.1.0.0
*ether2.ip-v4.tcp.hosts2-ns	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.81.4.0.1.0.0
*ether2.ip-v4.tcp.xfer	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.82.4.0.1.0.0
*ether2.ip-v4.tcp.mit-ml-dev	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.83.4.0.1.0.0
*ether2.ip-v4.tcp.ctf	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.84.4.0.1.0.0
*ether2.ip-v4.tcp.mit-ml-dev	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.85.4.0.1.0.0
*ether2.ip-v4.tcp.mfcobol	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.86.4.0.1.0.0
*ether2.ip-v4.tcp.priv-termlink	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.87.4.0.1.0.0
*ether2.ip-v4.tcp.kerberos	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.88.4.0.1.0.0
*ether2.ip-v4.tcp.su-mit-tg	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.89.4.0.1.0.0
*ether2.ip-v4.tcp.dnsix	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.90.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.mit-dov	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.91.4.0.1.0.0
*ether2.ip-v4.tcp.npp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.92.4.0.1.0.0
*ether2.ip-v4.tcp.dcp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.93.4.0.1.0.0
*ether2.ip-v4.tcp.objcall	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.94.4.0.1.0.0
*ether2.ip-v4.tcp.supdup	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.95.4.0.1.0.0
*ether2.ip-v4.tcp.dixie	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.96.4.0.1.0.0
*ether2.ip-v4.tcp.swift-rvf	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.97.4.0.1.0.0
*ether2.ip-v4.tcp.tacnews	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.98.4.0.1.0.0
*ether2.ip-v4.tcp.metagram	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.99.4.0.1.0.0
*ether2.ip-v4.tcp.newacct	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.100.4.0.1.0.0
*ether2.ip-v4.tcp.hostname	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.101.4.0.1.0.0
*ether2.ip-v4.tcp.iso-tsap	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.102.4.0.1.0.0
*ether2.ip-v4.tcp.gppitnp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.103.4.0.1.0.0
*ether2.ip-v4.tcp.acr-nema	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.104.4.0.1.0.0
*ether2.ip-v4.tcp.csnet-ns	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.105.4.0.1.0.0
*ether2.ip-v4.tcp.3com-tsmux	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.106.4.0.1.0.0
*ether2.ip-v4.tcp.rtelnet	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.107.4.0.1.0.0
*ether2.ip-v4.tcp.snagas	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.108.4.0.1.0.0
*ether2.ip-v4.tcp.pop2	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.109.4.0.1.0.0
*ether2.ip-v4.tcp.pop3	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.110.4.0.1.0.0
*ether2.ip-v4.tcp.sunrpc	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.111.4.0.1.0.0
*ether2.ip-v4.tcp.mcidas	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.112.4.0.1.0.0
*ether2.ip-v4.tcp.auth	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.113.4.0.1.0.0
*ether2.ip-v4.tcp.audionews	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.114.4.0.1.0.0
*ether2.ip-v4.tcp.sftp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.115.4.0.1.0.0
*ether2.ip-v4.tcp.ansanotify	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.116.4.0.1.0.0
*ether2.ip-v4.tcp.uucp-path	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.117.4.0.1.0.0
*ether2.ip-v4.tcp.sqlserv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.118.4.0.1.0.0
*ether2.ip-v4.tcp.nntp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.119.4.0.1.0.0
*ether2.ip-v4.tcp.erpc	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.121.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.smakynet	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.122.4.0.1.0.0
*ether2.ip-v4.tcp.ansatrader	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.124.4.0.1.0.0
*ether2.ip-v4.tcp.locus-map	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.125.4.0.1.0.0
*ether2.ip-v4.tcp.unitary	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.126.4.0.1.0.0
*ether2.ip-v4.tcp.locus-con	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.127.4.0.1.0.0
*ether2.ip-v4.tcp.gss-xlicen	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.128.4.0.1.0.0
*ether2.ip-v4.tcp.pwdgen	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.129.4.0.1.0.0
*ether2.ip-v4.tcp.cisco-fna	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.130.4.0.1.0.0
*ether2.ip-v4.tcp.cisco-tna	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.131.4.0.1.0.0
*ether2.ip-v4.tcp.cisco-sys	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.132.4.0.1.0.0
*ether2.ip-v4.tcp.statsrv	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.133.4.0.1.0.0
*ether2.ip-v4.tcp.ingres-net	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.134.4.0.1.0.0
*ether2.ip-v4.tcp.loc-srv	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.135.4.0.1.0.0
*ether2.ip-v4.tcp.profile	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.136.4.0.1.0.0
*ether2.ip-v4.tcp.netbios-ns	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.137.4.0.1.0.0
*ether2.ip-v4.tcp.netbios-dgm	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.138.4.0.1.0.0
*ether2.ip-v4.tcp.netbios-ssn	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.139.4.0.1.0.0
*ether2.ip-v4.tcp.emfis-data	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.140.4.0.1.0.0
*ether2.ip-v4.tcp.emfis-cntl	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.141.4.0.1.0.0
*ether2.ip-v4.tcp.bl-idm	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.142.4.0.1.0.0
*ether2.ip-v4.tcp.imap2	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.143.4.0.1.0.0
*ether2.ip-v4.tcp.news	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.144.4.0.1.0.0
*ether2.ip-v4.tcp.uaac	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.145.4.0.1.0.0
*ether2.ip-v4.tcp.iso-tp0	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.146.4.0.1.0.0
*ether2.ip-v4.tcp.iso-ip	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.147.4.0.1.0.0
*ether2.ip-v4.tcp.cronus	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.148.4.0.1.0.0
*ether2.ip-v4.tcp.aed-512	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.149.4.0.1.0.0
*ether2.ip-v4.tcp.sql-net	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.150.4.0.1.0.0
*ether2.ip-v4.tcp.hems	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.151.4.0.1.0.0
*ether2.ip-v4.tcp.bftp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.152.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.netsc-prod	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.154.4.0.1.0.0
*ether2.ip-v4.tcp.netsc-dev	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.155.4.0.1.0.0
*ether2.ip-v4.tcp.sqlsrv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.156.4.0.1.0.0
*ether2.ip-v4.tcp.knet-cmp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.157.4.0.1.0.0
*ether2.ip-v4.tcp.pcmail-srv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.158.4.0.1.0.0
*ether2.ip-v4.tcp.nss-routing	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.159.4.0.1.0.0
*ether2.ip-v4.tcp.snmp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.161.4.0.1.0.0
*ether2.ip-v4.tcp.snmptrap	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.162.4.0.1.0.0
*ether2.ip-v4.tcp.cmip-man	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.163.4.0.1.0.0
*ether2.ip-v4.tcp.cmip-agent	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.164.4.0.1.0.0
*ether2.ip-v4.tcp.xns-courier	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.165.4.0.1.0.0
*ether2.ip-v4.tcp.s-net	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.166.4.0.1.0.0
*ether2.ip-v4.tcp.namp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.167.4.0.1.0.0
*ether2.ip-v4.tcp.rsvd	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.168.4.0.1.0.0
*ether2.ip-v4.tcp.send	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.169.4.0.1.0.0
*ether2.ip-v4.tcp.print-srv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.170.4.0.1.0.0
*ether2.ip-v4.tcp.multiplex	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.171.4.0.1.0.0
*ether2.ip-v4.tcp.cl-1	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.172.4.0.1.0.0
*ether2.ip-v4.tcp.xyplex-mux	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.173.4.0.1.0.0
*ether2.ip-v4.tcp.mailq	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.174.4.0.1.0.0
*ether2.ip-v4.tcp.vmnet	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.175.4.0.1.0.0
*ether2.ip-v4.tcp.genrad-mux	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.176.4.0.1.0.0
*ether2.ip-v4.tcp.nextstep	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.178.4.0.1.0.0
*ether2.ip-v4.tcp.bgp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.179.4.0.1.0.0
*ether2.ip-v4.tcp.ris	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.180.4.0.1.0.0
*ether2.ip-v4.tcp.unify	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.181.4.0.1.0.0
*ether2.ip-v4.tcp.audit	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.182.4.0.1.0.0
*ether2.ip-v4.tcp.ocbinder	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.183.4.0.1.0.0
*ether2.ip-v4.tcp.ocserver	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.184.4.0.1.0.0
*ether2.ip-v4.tcp.remote-kis	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.185.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.kis	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.186.4.0.1.0.0
*ether2.ip-v4.tcp.aci	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.187.4.0.1.0.0
*ether2.ip-v4.tcp.mumps	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.188.4.0.1.0.0
*ether2.ip-v4.tcp.qft	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.189.4.0.1.0.0
*ether2.ip-v4.tcp.gacp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.190.4.0.1.0.0
*ether2.ip-v4.tcp.prospero	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.191.4.0.1.0.0
*ether2.ip-v4.tcp.osu-nms	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.192.4.0.1.0.0
*ether2.ip-v4.tcp.srmp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.193.4.0.1.0.0
*ether2.ip-v4.tcp.irc	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.194.4.0.1.0.0
*ether2.ip-v4.tcp.dn6-nlm-aud	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.195.4.0.1.0.0
*ether2.ip-v4.tcp.dn6-smm-red	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.196.4.0.1.0.0
*ether2.ip-v4.tcp.dls	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.197.4.0.1.0.0
*ether2.ip-v4.tcp.dls-mon	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.198.4.0.1.0.0
*ether2.ip-v4.tcp.smux	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.199.4.0.1.0.0
*ether2.ip-v4.tcp.src	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.200.4.0.1.0.0
*ether2.ip-v4.tcp.at-rtmp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.201.4.0.1.0.0
*ether2.ip-v4.tcp.at-nbp	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.202.4.0.1.0.0
*ether2.ip-v4.tcp.at-3	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.203.4.0.1.0.0
*ether2.ip-v4.tcp.at-echo	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.204.4.0.1.0.0
*ether2.ip-v4.tcp.at-5	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.205.4.0.1.0.0
*ether2.ip-v4.tcp.at-zis	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.206.4.0.1.0.0
*ether2.ip-v4.tcp.at-7	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.207.4.0.1.0.0
*ether2.ip-v4.tcp.at-8	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.208.4.0.1.0.0
*ether2.ip-v4.tcp.tam	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.209.4.0.1.0.0
*ether2.ip-v4.tcp.z39-50	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.210.4.0.1.0.0
*ether2.ip-v4.tcp.914c-g	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.211.4.0.1.0.0
*ether2.ip-v4.tcp.anet	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.212.4.0.1.0.0
*ether2.ip-v4.tcp.vmpwscs	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.214.4.0.1.0.0
*ether2.ip-v4.tcp.softpc	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.215.4.0.1.0.0
*ether2.ip-v4.tcp.atls	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.0.216.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.dbase	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.217.4.0.1.0.0
*ether2.ip-v4.tcp.mpp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.218.4.0.1.0.0
*ether2.ip-v4.tcp.uarps	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.219.4.0.1.0.0
*ether2.ip-v4.tcp.imap3	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.220.4.0.1.0.0
*ether2.ip-v4.tcp.fln-spx	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.221.4.0.1.0.0
*ether2.ip-v4.tcp.rsh-spx	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.222.4.0.1.0.0
*ether2.ip-v4.tcp.cdc	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.223.4.0.1.0.0
*ether2.ip-v4.tcp.sur-meas	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.243.4.0.1.0.0
*ether2.ip-v4.tcp.link	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.245.4.0.1.0.0
*ether2.ip-v4.tcp.dsp3270	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.0.246.4.0.1.0.0
*ether2.ip-v4.tcp.ldap	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.1.133.4.0.1.0.0
*ether2.ip-v4.tcp.https	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.1.187.4.0.1.0.0
*ether2.ip-v4.tcp.exec	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.0.4.0.1.0.0
*ether2.ip-v4.tcp.login	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.1.4.0.1.0.0
*ether2.ip-v4.tcp.cmd	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.2.4.0.1.0.0
*ether2.ip-v4.tcp.printer	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.3.4.0.1.0.0
*ether2.ip-v4.tcp.uucp	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.28.4.0.1.0.0
*ether2.ip-v4.tcp.banyan-vip	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.61.4.0.1.0.0
*ether2.ip-v4.tcp.doom	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.2.154.4.0.1.0.0
*ether2.ip-v4.tcp.notes	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.5.72.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-srv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.5.245.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-tns	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.5.246.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-tns-srv	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.5.247.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-coauthor	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.5.249.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-remdb	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.6.35.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-names	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.6.39.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-em1	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.6.212.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-em2	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.6.218.4.0.1.0.0
*ether2.ip-v4.tcp.ms-streaming	16.1.0.0.1.0.0.8.0.0.0.0.0.6.0.0.6.219.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.tcp.oracl-vp2	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.7.16.4.0.1.0.0
*ether2.ip-v4.tcp.oracl-vp1	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.7.17.4.0.1.0.0
*ether2.ip-v4.tcp.ccmail	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.12.192.4.0.1.0.0
*ether2.ip-v4.tcp.xwin	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.23.112.4.0.1.0.0
*ether2.ip-v4.tcp.quake	16.1.0.0.1.0.0.8.0.0.0.0.6.0.0.101.144.4.0.1.0.0
UDP Applications	
*ether2.ip-v4.udp.echo	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.7.4.0.1.0.0
*ether2.ip-v4.udp.discard	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.9.4.0.1.0.0
*ether2.ip-v4.udp.systat	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.11.4.0.1.0.0
*ether2.ip-v4.udp.daytime	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.13.4.0.1.0.0
*ether2.ip-v4.udp.qotd	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.17.4.0.1.0.0
*ether2.ip-v4.udp.msp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.18.4.0.1.0.0
*ether2.ip-v4.udp.chargen	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.19.4.0.1.0.0
*ether2.ip-v4.udp.priv-mail	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.24.4.0.1.0.0
*ether2.ip-v4.udp.nsw-fe	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.27.4.0.1.0.0
*ether2.ip-v4.udp.msg-icp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.29.4.0.1.0.0
*ether2.ip-v4.udp.msg-auth	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.31.4.0.1.0.0
*ether2.ip-v4.udp.dsp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.33.4.0.1.0.0
*ether2.ip-v4.udp.priv-print	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.35.4.0.1.0.0
*ether2.ip-v4.udp.time	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.37.4.0.1.0.0
*ether2.ip-v4.udp.rlp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.39.4.0.1.0.0
*ether2.ip-v4.udp.graphics	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.41.4.0.1.0.0
*ether2.ip-v4.udp.nameserver	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.42.4.0.1.0.0
*ether2.ip-v4.udp.auditd	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.48.4.0.1.0.0
*ether2.ip-v4.udp.re-mail-ck	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.50.4.0.1.0.0
*ether2.ip-v4.udp.la-maint	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.51.4.0.1.0.0
*ether2.ip-v4.udp.xns-time	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.52.4.0.1.0.0
*ether2.ip-v4.udp.domain	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.53.4.0.1.0.0
*ether2.ip-v4.udp.xns-ch	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.54.4.0.1.0.0
*ether2.ip-v4.udp.isi-gl	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.55.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.udp.xns-auth	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.56.4.0.1.0.0
*ether2.ip-v4.udp.priv-term	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.57.4.0.1.0.0
*ether2.ip-v4.udp.xns-mail	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.58.4.0.1.0.0
*ether2.ip-v4.udp.priv-file	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.59.4.0.1.0.0
*ether2.ip-v4.udp.ni-mail	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.61.4.0.1.0.0
*ether2.ip-v4.udp.bootps	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.67.4.0.1.0.0
*ether2.ip-v4.udp.bootpc	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.68.4.0.1.0.0
*ether2.ip-v4.udp.tftp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.69.4.0.1.0.0
*ether2.ip-v4.udp.priv-dialout	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.75.4.0.1.0.0
*ether2.ip-v4.udp.deos	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.76.4.0.1.0.0
*ether2.ip-v4.udp.priv-rje	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.77.4.0.1.0.0
*ether2.ip-v4.udp.vettcp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.78.4.0.1.0.0
*ether2.ip-v4.udp.hosts2-ns	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.81.4.0.1.0.0
*ether2.ip-v4.udp.xfer	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.82.4.0.1.0.0
*ether2.ip-v4.udp.mit-ml-dev	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.83.4.0.1.0.0
*ether2.ip-v4.udp.ctf	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.84.4.0.1.0.0
*ether2.ip-v4.udp.mit-ml-dev	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.85.4.0.1.0.0
*ether2.ip-v4.udp.kerberos	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.88.4.0.1.0.0
*ether2.ip-v4.udp.npp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.92.4.0.1.0.0
*ether2.ip-v4.udp.dcp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.93.4.0.1.0.0
*ether2.ip-v4.udp.dixie	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.96.4.0.1.0.0
*ether2.ip-v4.udp.swift-rvf	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.97.4.0.1.0.0
*ether2.ip-v4.udp.tacnews	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.98.4.0.1.0.0
*ether2.ip-v4.udp.metagram	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.99.4.0.1.0.0
*ether2.ip-v4.udp.iso-tsap	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.102.4.0.1.0.0
*ether2.ip-v4.udp.gppitnp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.103.4.0.1.0.0
*ether2.ip-v4.udp.csnet-ns	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.105.4.0.1.0.0
*ether2.ip-v4.udp.3com-tsmux	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.106.4.0.1.0.0
*ether2.ip-v4.udp.pop3	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.110.4.0.1.0.0
*ether2.ip-v4.udp.sunrpc	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.111.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.udp.audionews	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.114.4.0.1.0.0
*ether2.ip-v4.udp.ansanotify	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.116.4.0.1.0.0
*ether2.ip-v4.udp.sqlserv	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.118.4.0.1.0.0
*ether2.ip-v4.udp.cfdptkt	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.120.4.0.1.0.0
*ether2.ip-v4.udp.erpc	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.121.4.0.1.0.0
*ether2.ip-v4.udp.smakynet	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.122.4.0.1.0.0
*ether2.ip-v4.udp.ntp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.123.4.0.1.0.0
*ether2.ip-v4.udp.ansatrader	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.124.4.0.1.0.0
*ether2.ip-v4.udp.unitary	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.126.4.0.1.0.0
*ether2.ip-v4.udp.gss-xlicen	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.128.4.0.1.0.0
*ether2.ip-v4.udp.pwdgen	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.129.4.0.1.0.0
*ether2.ip-v4.udp.cisco-fna	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.130.4.0.1.0.0
*ether2.ip-v4.udp.cisco-tna	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.131.4.0.1.0.0
*ether2.ip-v4.udp.cisco-sys	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.132.4.0.1.0.0
*ether2.ip-v4.udp.statsrv	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.133.4.0.1.0.0
*ether2.ip-v4.udp.loc-srv	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.135.4.0.1.0.0
*ether2.ip-v4.udp.netbios-ns	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.137.4.0.1.0.0
*ether2.ip-v4.udp.netbios-dgm	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.138.4.0.1.0.0
*ether2.ip-v4.udp.netbios-ssn	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.139.4.0.1.0.0
*ether2.ip-v4.udp.emfis-data	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.140.4.0.1.0.0
*ether2.ip-v4.udp.emfis-cntl	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.141.4.0.1.0.0
*ether2.ip-v4.udp.bl-idm	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.142.4.0.1.0.0
*ether2.ip-v4.udp.news	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.144.4.0.1.0.0
*ether2.ip-v4.udp.uaac	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.145.4.0.1.0.0
*ether2.ip-v4.udp.iso-tp0	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.146.4.0.1.0.0
*ether2.ip-v4.udp.iso-ip	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.147.4.0.1.0.0
*ether2.ip-v4.udp.cronus	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.148.4.0.1.0.0
*ether2.ip-v4.udp.aed-512	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.149.4.0.1.0.0
*ether2.ip-v4.udp.sql-net	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.150.4.0.1.0.0
*ether2.ip-v4.udp.sgmp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.153.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.udp.netsc-prod	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.154.4.0.1.0.0
*ether2.ip-v4.udp.netsc-dev	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.155.4.0.1.0.0
*ether2.ip-v4.udp.nss-routing	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.159.4.0.1.0.0
*ether2.ip-v4.udp.sgmp-traps	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.160.4.0.1.0.0
*ether2.ip-v4.udp.snmp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.161.4.0.1.0.0
*ether2.ip-v4.udp.snmptrap	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.162.4.0.1.0.0
*ether2.ip-v4.udp.cmip-man	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.163.4.0.1.0.0
*ether2.ip-v4.udp.cmip-agent	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.164.4.0.1.0.0
*ether2.ip-v4.udp.xns-courier	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.165.4.0.1.0.0
*ether2.ip-v4.udp.s-net	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.166.4.0.1.0.0
*ether2.ip-v4.udp.namp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.167.4.0.1.0.0
*ether2.ip-v4.udp.rsvd	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.168.4.0.1.0.0
*ether2.ip-v4.udp.send	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.169.4.0.1.0.0
*ether2.ip-v4.udp.print-srv	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.170.4.0.1.0.0
*ether2.ip-v4.udp.multiplex	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.171.4.0.1.0.0
*ether2.ip-v4.udp.cl-1	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.172.4.0.1.0.0
*ether2.ip-v4.udp.xyplex-mux	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.173.4.0.1.0.0
*ether2.ip-v4.udp.mailq	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.174.4.0.1.0.0
*ether2.ip-v4.udp.vmnet	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.175.4.0.1.0.0
*ether2.ip-v4.udp.genrad-mux	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.176.4.0.1.0.0
*ether2.ip-v4.udp.xdmcp	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.177.4.0.1.0.0
*ether2.ip-v4.udp.nextstep	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.178.4.0.1.0.0
*ether2.ip-v4.udp.ris	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.180.4.0.1.0.0
*ether2.ip-v4.udp.unify	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.181.4.0.1.0.0
*ether2.ip-v4.udp.audit	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.182.4.0.1.0.0
*ether2.ip-v4.udp.ocbinder	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.183.4.0.1.0.0
*ether2.ip-v4.udp.ocserver	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.184.4.0.1.0.0
*ether2.ip-v4.udp.remote-kis	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.185.4.0.1.0.0
*ether2.ip-v4.udp.kis	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.186.4.0.1.0.0
*ether2.ip-v4.udp.aci	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.0.187.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.udp.mumps	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.188.4.0.1.0.0
*ether2.ip-v4.udp.osu-nms	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.192.4.0.1.0.0
*ether2.ip-v4.udp.srmp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.193.4.0.1.0.0
*ether2.ip-v4.udp.irc	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.194.4.0.1.0.0
*ether2.ip-v4.udp.dls	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.197.4.0.1.0.0
*ether2.ip-v4.udp.dls-mon	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.198.4.0.1.0.0
*ether2.ip-v4.udp.src	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.200.4.0.1.0.0
*ether2.ip-v4.udp.at-rtmp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.201.4.0.1.0.0
*ether2.ip-v4.udp.at-nbp	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.202.4.0.1.0.0
*ether2.ip-v4.udp.at-3	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.203.4.0.1.0.0
*ether2.ip-v4.udp.at-echo	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.204.4.0.1.0.0
*ether2.ip-v4.udp.at-5	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.205.4.0.1.0.0
*ether2.ip-v4.udp.at-zis	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.206.4.0.1.0.0
*ether2.ip-v4.udp.at-7	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.207.4.0.1.0.0
*ether2.ip-v4.udp.at-8	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.208.4.0.1.0.0
*ether2.ip-v4.udp.tam	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.209.4.0.1.0.0
*ether2.ip-v4.udp.914c-g	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.211.4.0.1.0.0
*ether2.ip-v4.udp.anet	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.212.4.0.1.0.0
*ether2.ip-v4.udp.ipx-tunnel	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.213.4.0.1.0.0
*ether2.ip-v4.udp.vmpwscs	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.214.4.0.1.0.0
*ether2.ip-v4.udp.softpc	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.215.4.0.1.0.0
*ether2.ip-v4.udp.atls	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.216.4.0.1.0.0
*ether2.ip-v4.udp.dbase	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.217.4.0.1.0.0
*ether2.ip-v4.udp.uarps	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.219.4.0.1.0.0
*ether2.ip-v4.udp.fln-spx	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.221.4.0.1.0.0
*ether2.ip-v4.udp.rsh-spx	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.222.4.0.1.0.0
*ether2.ip-v4.udp.cdc	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.223.4.0.1.0.0
*ether2.ip-v4.udp.sur-meas	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.243.4.0.1.0.0
*ether2.ip-v4.udp.link	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.245.4.0.1.0.0
*ether2.ip-v4.udp.dsp3270	16.1.0.0.1.0.0.8.0.0.0.0.17.0.0.0.246.4.0.1.0.0

Protocol Encapsulation	Protocol Identifier (Object ID)
*ether2.ip-v4.udp.ldap	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.1.133.4.0.1.0.0
*ether2.ip-v4.udp.biff	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.2.0.4.0.1.0.0
*ether2.ip-v4.udp.who	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.2.1.4.0.1.0.0
*ether2.ip-v4.udp.syslog	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.2.2.4.0.1.0.0
*ether2.ip-v4.udp.ip-xns-rip	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.2.8.4.0.1.0.0
*ether2.ip-v4.udp.banyan-vip	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.2.61.4.0.1.0.0
*ether2.ip-v4.udp.notes	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.5.72.4.0.1.0.0
*ether2.ip-v4.udp.ccmail	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.12.192.4.0.1.0.0
*ether2.ip-v4.udp.quake	16.1.0.0.1.0.0.8.0.0.0.0.0.17.0.0.101.144.4.0.1.0.0