JUNOSe™ Software
for E-series™ Routing Platforms

IP, IPv6, and IGP Configuration Guide

Release 9.0.x

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 162-01844-00, Revision A00

Juniper Networks, the Juniper Networks logo, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. JUNOS and JUNOSe are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks (including the ERX-310, ERX-705, ERX-710, ERX-1410, ERX-1440, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, and T320 routers, T640 routing node, and the JUNOS, JUNOSe, and SDX-300 software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Copyright © 2008, Juniper Networks, Inc.
All rights reserved. Printed in USA.

JUNOSe™ Software for E-series™ Routing Platforms IP, IPv6, and IGP Configuration Guide, Release 9.0.x
Writing: Mark Barnard, Bruce Gillham, Sarah Lesway-Ball, Brian Wesley Simmons, Fran Singer
Editing: Ben Mann, Fran Mues
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
29 February 2008—Revision 1
The information in this document is current as of the date listed in the revision history.

Software License

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper equipment.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use the Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius software on multiple computers requires multiple licenses, regardless of whether such computers are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Embedded Software on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Interface Information. To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. Third Party Software. Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

About This Guide

This preface provides the following guidelines for using *JUNOSe IP, IPv6, and IGP Configuration Guide*:

Objectives

This guide provides the information you need to configure certain routing protocols (primarily IP and IPv6 protocols) on your E-series router.

An E-series router is shipped with the latest system software installed. If you need to install a future release or reinstall the system software, refer to the procedures in *JUNOSe System Basics Configuration Guide, Chapter 3, Installing JUNOSe Software*.

NOTE: If the information in the latest *JUNOSe Release Notes* differs from the information in this guide, follow the *JUNOSe Release Notes*.

Audience

This guide is intended for experienced system and network specialists working with E-series routers in an Internet access environment.

E-series Routers

Seven models of E-series routers are available:

- E120 router
- E320 router
- ERX-1440 router
- ERX-1410 router
- ERX-710 router
- ERX-705 router
- ERX-310 router

All models use the same software. For information about all models except the E120 router and the E320 router, see *ERX Hardware Guide, Chapter 1, ERX Overview*. For information about the E120 router and the E320 router, see *E120 and E320 Hardware Guide*, *Chapter 1*, *E120 and E320 Overview*.

In the E-series documentation, the term ERX-14xx models refers to both the ERX-1440 router and the ERX-1410 router. Similarly, the term ERX-7xx models refers to both the ERX-710 router and the ERX-705 router. The terms ERX-1440 router, ERX-1410 router, ERX-710 router, ERX-705 router, ERX-310 router, E120 router, and E320 router refer to the specific models.

Documentation Conventions

Table 1 defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury.

Table 2 defines text conventions used in this guide and the syntax conventions used primarily in the *JUNOSe Command Reference Guide*. For more information about command syntax, see *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Text Conventions		
Bold text like this	Represents commands and keywords in text.	■ Issue the **clock source** command. ■ Specify the keyword **exp-msg**.
Bold text like this	Represents text that the user must type.	host1(config)#**traffic class low-loss1**

`Fixed-width text like this`	Represents information as displayed on your terminal's screen.	`host1#show ip ospf 2`
		`Routing Process OSPF 2 with Router ID 5.5.0.250`
		`Router is an Area Border Router (ABR)`

Italic text like this	■ Emphasizes words. ■ Identifies variables. ■ Identifies chapter, appendix, and book names.	■ There are two levels of access, *user* and *privileged*. ■ *clusterId*, *ipAddress*. ■ *Appendix A*, *System Specifications*.
Plus sign (+) linking key names	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Syntax Conventions in the *Command Reference Guide*		
Plain text like this	Represents keywords.	terminal length
Italic text like this	Represents variables.	*mask*, *accessListName*
| (pipe symbol)	Represents a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable can be either optional or required.)	diagnostic | line
[ ] (brackets)	Represent optional keywords or variables.	[internal | external]
[ ]* (brackets and asterisk)	Represent optional keywords or variables that can be entered more than once.	[level1 | level2 | l1]*
{ } (braces)	Represent required keywords or variables.	{ permit | deny } { in | out } { *clusterId* | *ipAddress* }

Related E-series and JUNOSe Documentation

The E-series and JUNOSe documentation set consists of several hardware and software guides, which are available in electronic and printed formats.

E-series and JUNOSe Documents

Table 3 lists and describes the E-series and JUNOSe document set. For a complete list of abbreviations used in this document set, along with their spelled-out terms, see *JUNOSe System Basics Configuration Guide, Appendix A, Abbreviations and Acronyms*.

Table 3: Juniper Networks E-series and JUNOSe Technical Publications

Document	Description
E-series Hardware Documentation	
E120 and E320 Quick Start Guide	Shipped in the box with all new E120 and E320 routers. Provides the basic procedures to help you get the routers up and running quickly.
E120 and E320 Hardware Guide	Provides the necessary procedures for getting E120 routers and E320 routers operational, including information about: ■ Installing the chassis and modules ■ Connecting cables ■ Powering up the routers ■ Configuring the routers for management access ■ Troubleshooting common issues Describes switch route processor (SRP) modules, line modules, and I/O adapters (IOAs) available for E120 and E320 routers.
E120 and E320 Module Guide	Provides detailed specifications for line modules and IOAs in E120 and E320 routers, and information about the compatibility of these modules with JUNOSe software releases. Lists the layer 2 protocols, layer 3 protocols, and applications that line modules and their corresponding IOAs support. Provides module LED information.
E-series Installation Quick Start poster or *ERX Quick Start Guide*	Shipped in the box with all new ERX routers. Provides the basic procedures to help you get an ERX router up and running quickly.
ERX Hardware Guide	Provides the necessary procedures for getting ERX-14xx models, ERX-7xx models, and ERX-310 routers operational, including information about: ■ Installing the chassis and modules ■ Connecting cables ■ Powering up the routers ■ Configuring the routers for management access ■ Troubleshooting common issues Describes switch route processor (SRP) modules, line modules, and I/O modules available for the ERX routers.
ERX Module Guide	Provides detailed specifications for line modules and I/O modules in ERX-14xx models, ERX-7xx models, and ERX-310 routers, and information about the compatibility of these modules with JUNOSe software releases. Lists the layer 2 protocols, layer 3 protocols, and applications that line modules and their corresponding I/O modules support. Provides module LED information.

Table 3: Juniper Networks E-series and JUNOSe Technical Publications (continued)

Document	Description
ERX End-of-Life Module Guide	Provides an overview and description of ERX modules that are end-of-life (EOL) and can no longer be ordered for the following routers: ■ ERX-7xx models ■ ERX-14xx models ■ ERX-310 router
JUNOSe Software Guides	
JUNOSe System Basics Configuration Guide	Provides information about: ■ Planning and configuring your network ■ Using the command-line interface (CLI) ■ Installing JUNOSe software ■ Configuring the Simple Network Management Protocol (SNMP) ■ Managing the router and its modules, including the use of high availability (HA) for SRP redundancy ■ Configuring and running a unified in-service software upgrade (ISSU) ■ Configuring passwords and security ■ Configuring the router clock ■ Configuring virtual routers
JUNOSe Physical Layer Configuration Guide	Explains how to configure, test, and monitor physical layer interfaces.
JUNOSe Link Layer Configuration Guide	Explains how to configure and monitor static and dynamic link layer interfaces.
JUNOSe IP, IPv6, and IGP Configuration Guide	Explains how to configure and monitor IP, IPv6 and Neighbor Discovery, and interior gateway protocols (RIP, OSPF, and IS-IS).
JUNOSe IP Services Configuration Guide	Explains how to configure and monitor IP routing services. Topics include: ■ Routing policies ■ Firewalls ■ Network Address Translation (NAT) ■ J-Flow statistics ■ Bidirectional forwarding detection (BFD) ■ Internet Protocol Security (IPSec) ■ Access Node Control Protocol (ANCP), also known as Layer 2 Control (L2C) ■ Digital certificates ■ IP tunnels ■ Virtual Router Redundancy Protocol (VRRP) ■ Mobile IP home agent
JUNOSe Multicast Routing Configuration Guide	Explains how to configure and monitor IP multicast routing and IPv6 multicast routing. Topics include: ■ Internet Group Management Protocol (IGMP) ■ Protocol Independent Multicast (PIM) ■ Distance Vector Multicast Routing Protocol (DVMRP) ■ Multicast Listener Discovery (MLD)

Table 3: Juniper Networks E-series and JUNOSe Technical Publications (continued)

Document	Description
JUNOSe BGP and MPLS Configuration Guide	Explains how to configure and monitor: ■ Border Gateway Protocol (BGP) routing ■ Multiprotocol Label Switching (MPLS) and related applications ■ Layer 2 services over MPLS ■ Virtual private LAN service (VPLS) ■ Layer 2 virtual private networks (L2VPNs)
JUNOSe Policy Management Configuration Guide	Explains how to configure, manage, and monitor customized policy rules for packet classification, forwarding, filtering, and flow rates. Also describes the packet mirroring feature, which uses secure policies.
JUNOSe Quality of Service Configuration Guide	Explains how to configure quality of service (QoS) features to queue, schedule, and monitor traffic flow. These features include: ■ Traffic classes and traffic-class groups ■ Drop, queue, QoS, and scheduler profiles ■ QoS parameters ■ Statistics
JUNOSe Broadband Access Configuration Guide	Explains how to configure and monitor a remote access environment, which can include the following features: ■ Authentication, authorization, and accounting (AAA) ■ Dynamic Host Configuration Protocol (DHCP) ■ Remote Authentication Dial-In User Service (RADIUS) ■ Terminal Access Controller Access Control System (TACACS+) ■ Layer 2 Tunneling Protocol (L2TP) ■ Subscriber management
JUNOSe System Event Logging Reference Guide	Describes the JUNOSe system logging feature and describes how to use the CLI to monitor your system's log configuration and system events.
JUNOSe Command Reference Guide A to M; JUNOSe Command Reference Guide N to Z	Together constitute the *JUNOSe Command Reference Guide*. Contain important information about commands implemented in the system software. Use to look up: ■ Descriptions of commands and command parameters ■ Command syntax ■ A command's related mode ■ Starting with JUNOSe Release 7.1.0, a history of when a command, its keywords, and its variables were introduced or added Use with the JUNOSe configuration guides.
JUNOSe Comprehensive Index	Provides a complete index of the JUNOSe software documentation set.
JUNOSe Glossary	Provides definitions for terms used in JUNOSe technical documentation.
Release Notes	
JUNOSe Release Notes	Provide the latest information about features, changes, known problems, resolved problems, and system maximum values. If the information in the *Release Notes* differs from the information found in the documentation set, follow the *Release Notes*. Release notes are included on the corresponding software CD and are available on the Web.

JUNOSe Configuration Guides

JUNOSe software configuration guides use a bottom-up approach to describe the relationship of layers, protocols, and interfaces in the configuration process. For more information, see *Layered Approach* in *JUNOSe System Basics Configuration Guide, Chapter 1, Planning Your Network*.

The chapters in JUNOSe software configuration guides typically include the following topics:

- Conceptual and overview information
- Information you need to know or tasks you need to perform before you begin
- Platform-specific issues you need to take into consideration
- Applicable references, such as RFCs and IETF draft documents, about the protocols and features supported by the router
- Required and optional tasks, as step-by-step procedures
- Descriptions and examples of the commands you use
- Illustrations of network topologies
- Examples of command sequences for configuration, testing, and monitoring activities
- Sample displays that result when you issue the **show** command

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the products documentation page on the Juniper Networks Web site at http://www.juniper.net/.

To order printed copies of this manual and other Juniper Networks technical documents or to order a documentation CD, which contains this manual, contact your sales representative.

Copies of the Management Information Bases (MIBs) available in a software release are included on the software CDs and at http://www.juniper.net/.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at http://www.juniper.net/techpubs/docbug/docbugreport.html. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- **JTAC Policies**—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/customers/support/downloads/710059.pdf
- **Product Warranties**—For product warranty information, visit http://www.juniper.net/support/warranty/
- **JTAC Hours of Operation**—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/

- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Manager: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool located at https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Manager tool in the CSC at http://www.juniper.net/cm/
- Call 1-888-314-JTAC (1-888-314-5822 – toll free in the USA, Canada, and Mexico)

For international or direct-dial options in countries without toll-free numbers, visit http://www.juniper.net/support/requesting-support.html

Part 1

Internet Protocol

Chapter 1

Configuring IP

This chapter describes how to configure Internet Protocol (IP) routing on your E-series router.

Overview

TCP/IP is a suite of data communications protocols. Two of the more important protocols in the suite are the Transmission Control Protocol (TCP) and the Internet Protocol (IP).

IP provides the basic packet delivery service for all TCP/IP networks. IP is a *connectionless* protocol, which means that it does not exchange control information to establish an end-to-end connection before transmitting data. A *connection-oriented* protocol exchanges control information with the remote computer to verify that it is ready to receive data before sending it.

IP relies on protocols in other layers to establish the connection if connection-oriented services are required and to provide error detection and error recovery. IP is sometimes called an unreliable protocol, because it contains no error detection or recovery code.

IP Packets

A *packet* is a block of data that carries with it the information necessary to deliver it to a destination address. A *packet-switching network* uses the addressing information in the packets to switch packets from one physical network to another, moving them toward their final destination. Each packet travels the network independently of any other packet. The *datagram* is the packet format defined by IP.

IP Functions

Some of the functions IP performs include:

- Moving data between the network access layer and the host-to-host transport layer
- Routing datagrams to remote hosts
- Fragmenting and reassembling datagrams

Moving Data Between Layers

When IP receives a datagram that is addressed to the local host, it must pass the data portion of the datagram to the correct host-to-host transport layer protocol. IP uses the *protocol number* in the datagram header to select the transport layer protocol. Each host-to-host transport layer protocol has a unique protocol number that identifies it to IP.

Routing Datagrams to Remote Hosts

Internet gateways are commonly referred to as IP routers because they use IP to route packets between networks. In traditional TCP/IP terms, there are only two types of network devices: gateways and hosts. Gateways forward packets between networks, and hosts do not. However, if a host is connected to more than one network (called a *multihomed host*), it can forward packets between the networks. When a multihomed host forwards packets, it acts like any other gateway and is considered to be a gateway.

Fragmenting and Reassembling Datagrams

As a datagram is routed through different networks, it may be necessary for the IP module in a gateway to divide the datagram into smaller pieces. A datagram received from one network may be too large to be transmitted in a single packet on a different network. This condition occurs only when a gateway interconnects dissimilar physical networks.

Each type of network has a maximum transmission unit (MTU) that determines the largest packet it can transfer. If the datagram received from one network is longer than the other network's MTU, it is necessary to divide the datagram into smaller fragments for transmission in a process called *fragmentation*. See *Fragmentation* later in this chapter.

IP Layering

TCP/IP is organized into four conceptual layers (as shown in Figure 1).

Figure 1: TCP/IP Conceptual Layers

Network Interface Layer

The network interface layer is the lowest level of the TCP/IP protocol stack. It is responsible for transmitting datagrams over the physical medium to their final destinations.

Internet Layer

The Internet layer is the second level of the TCP/IP protocol stack. It provides host-to-host communication. In this layer, packets are encapsulated into datagrams, routing algorithms are run, and the datagram is passed to the network interface layer for transmission on the attached network.

Transport Layer

The transport layer is the third level of the TCP/IP protocol stack. It is responsible for providing communication between applications residing in different hosts. By placing identifying information in the datagram (such as socket information), the transport layer enables process-to-process communication.

The transport layer provides either a reliable transport service (TCP) or an unreliable service (User Data Protocol). In a reliable delivery service, the destination station acknowledges the receipt of a datagram.

Application Layer

The application layer is the fourth and highest level of the TCP/IP protocol stack. Some applications that run in this layer are:

- Telnet
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Simple Network Management Protocol (SNMP)
- Domain Name System (DNS)

Platform Considerations

For information about modules that support IP on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IP.

For information about modules that support IP on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support IP.

References

For more information about IP, consult the following resources:

- RFC 768—User Datagram Protocol (August 1980)
- RFC 791—Internet Protocol DARPA Internet Program Protocol Specification (September 1981)
- RFC 792—Internet Control Message Protocol (September 1981)
- RFC 793—Transmission Control Protocol (September 1981)
- RFC 854—Telnet Protocol Specification (May 1983)
- RFC 919—Broadcasting Internet Datagrams (October 1984)
- RFC 922—Broadcasting Internet Datagrams in the Presence of Subnets (October 1984)

- RFC 950—Internet Standard Subnetting Procedure (August 1985)
- RFC 1112—Host Extensions for IP Multicasting (August 1989)
- RFC 1122—Requirements for Internet Hosts—Communication Layers (October 1989)
- RFC 1812—Requirements for IP Version 4 Routers (June 1995)
- RFC 3419—Textual Conventions for Transport Addresses (December 2002)
- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.

IP Features

The E-series router supports the following IP features:

- Internet Control Message Protocol (ICMP)
- Traceroute
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Classless interdomain routing (CIDR)
- Maximum transmission unit (MTU)
- Support for simultaneous multiple logical IP stacks on the same router
- Flexible IP address assignment to support any portion of a physical interface (for example, a channel or circuit), exactly one physical interface, or multilink PPP interfaces
- Packet segmentation and reassembly
- Loose source routing to specify the IP route
- Strict source routing to specify the IP route for each hop
- Record route to track the route taken
- Internet timestamp
- Broadcast addressing, both limited broadcast and directed broadcast
- Support for 32,000 discrete, simultaneous IP interfaces per router to support thousands of logical connections
- Capability of detecting and reporting changes in the up or down state of any IP interface

- IP policy support. See *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*, for more information about policy configuration.
- Indirect next hops
- IP tunnel routing tables

IP Addressing

This section provides an overview of IP addressing in general and includes a discussion of CIDR, which your router fully supports.

Physical and Logical Addresses

Physical node addresses are used at the network access layer to identify physical devices in a network. For example, each Ethernet controller comes from the manufacturer with a physical address, called a MAC address.

IP implements a system of logical host addresses called IP addresses. The IP addresses are used by the internetwork and higher layers to identify devices and to perform internetwork routing. The Address Resolution Protocol (ARP) enables IP to identify the physical (MAC) address that matches a given IP address. You can use ARP only on Ethernet and bridged IP 1483 interfaces.

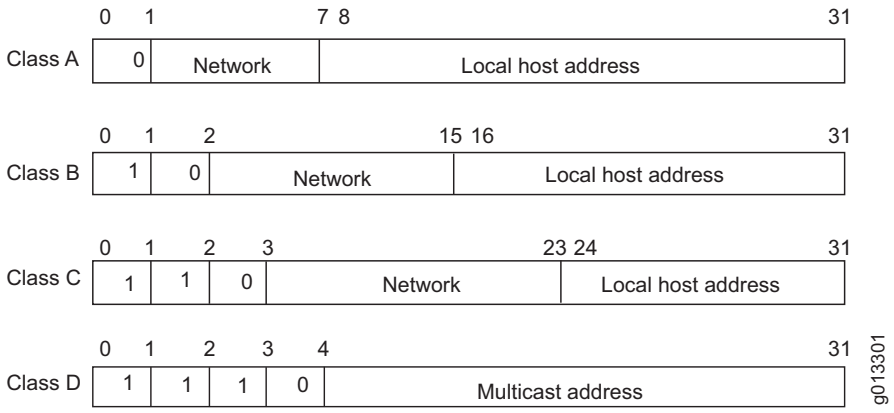
IP is used by all protocols in the layers above and below it to deliver data. This means that all TCP/IP data flows through IP when it is sent and received, regardless of its final destination.

Internet Addresses

Internet addressing uses a 32-bit address field. The bits in this address field are numbered 0 to 31. The 32-bit address field consists of two parts: a network number and a host number whose boundaries are defined based on the class of IP address. Hosts attached to the same network must share a common prefix designating their network number.

Four types of IP classes lend themselves to different network configurations, depending on the desired ratio of networks to hosts. Figure 2 shows the format of IP address classes.

Figure 2: IP Address Classes



- Class A—The leading bit is set to 0, a 7-bit number, and a 24-bit local host address. Up to 125 class A networks can be defined, with up to 16,777,214 hosts per network.
- Class B—The two highest-order bits are set to 1 and 0, a 14-bit network number, and a 16-bit local host address. Up to 16,382 class B networks can be defined, with up to 65,534 hosts per network.
- Class C—The three leading bits are set to 1, 1, and 0, a 21-bit network number, and an 8-bit local host address. Up to 2,097,152 class C networks can be defined, with up to 254 hosts per network.
- Class D—The four highest-order bits are set to 1, 1, 1, and 0. Class D is used as a multicast address.

Subnetwork Mask Format Options

Most commands allow you to specify IPv4 subnetwork masks in one of two ways: dotted decimal or prefix length notation.

NOTE: Protocol commands that use a reverse mask format (for example, RIP) cannot use the prefix notation format. Use the CLI help to verify if a command supports the /N prefix notation.

Dotted decimal notation expresses IP addresses and masks in dotted quads - four octets separated by dots (A.B.C.D). In this format, each octet in the address or mask is represented as a decimal number and the dots are used as octet separators.

For example, an IP address and subnetwork mask in dotted decimal notation would appear as follows:

10.10.24.6 255.255.0.0

Prefix length notation (often called network prefix format) allows for more efficient allocation of IP addresses than the old Class A, B, and C address scheme. The prefix length is the number of leftmost contiguous bits equal to 1 in the subnetwork mask. This format appears immediately following the dotted decimal IP address using a /N format.

NOTE: You can issue the network prefix with or without a space between the IP address and the network prefix (/N).

For example, the same IP address and subnetwork mask mentioned above would appear as follows using /N format:

10.10.24.6/16
or
10.10.24.6 /16

The following sections describe each subnetwork mask addressing method in more detail.

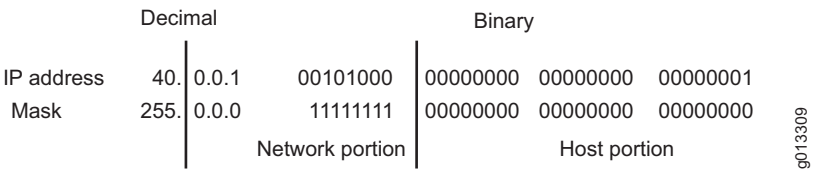
Subnet Addressing

A subnet is a subset of a class A, B, or C network. Subnets cannot be used with class D (multicast) addresses.

A network mask is used to separate the network information from the host information about an IP address. Figure 3 shows the network mask 255.0.0.0 applied to network 10.0.0.0. The mask in binary notation is a series of 1s followed by a series of contiguous 0s. The 1s represent the network number; the 0s represent the host number. The sample address splits the IP address 10.0.0.1 into a network portion of 10 and a host portion of 0.0.1.

NOTE: The router supports a 31-bit mask on point-to-point links. This means that the IP address 1.2.3.4 255.255.255.254 is valid. A point-to-point link in which only one end supports the use of 31-bit prefixes may not operate correctly.

Figure 3: Basic Network Masking



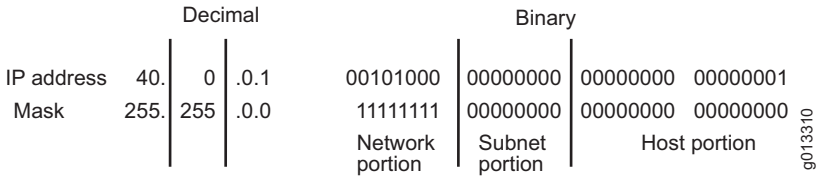
Classes A, B, and C have the following *natural masks,* which define the network and host portions of each class:

- Class A natural mask 255.0.0.
- Class B natural mask 255.255.0.0
- Class C natural mask 255.255.255.0

The use of masks can divide networks into subnetworks by extending the network portion of the address into the host portion. Subnetting increases the number of subnetworks and reduces the number of hosts.

For example, a network of the form 10.0.0.0 accommodates one physical segment with about 16 million hosts on it. Figure 4 shows how the mask 255.255.0.0. is applied to network 10.0.0.0. The mask divides the IP address 10.0.0.1 into a network portion of 10, a subnet portion of 0, and a host portion of 0.1. The mask has borrowed a portion of the host space and has applied it to the network space. The network space of the class 10 has increased from a single network 10.0.0.0 to 256 subnetworks, ranging from 10.0.0.0 to 10.255.0.0. This process decreases the number of hosts per subnet from 16,777,216 to 65,536.

Figure 4: Subnetting



Classless Addressing with CIDR

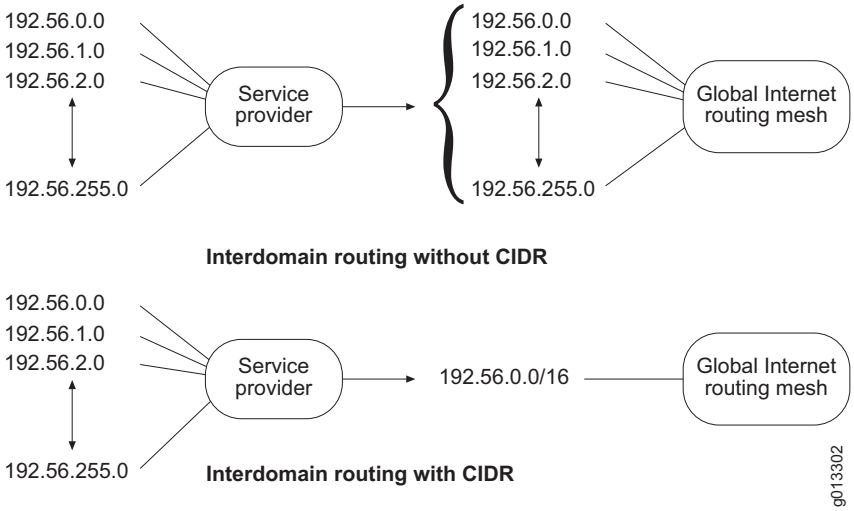
Classless interdomain routing (CIDR) is a system of addressing that improves the scaling factor of routing in the Internet. CIDR does not use an implicit mask based on the class of network. In CIDR, an IP network is represented by a prefix, which is an IP address and an indication of the leftmost contiguous significant bits within this address.

For example, without CIDR, the class C network address 192.56.0.0 would be an illegal address. With CIDR, the address becomes valid with the notation: 192.56.0.0/16. The /16 indicates that 16 bits of mask are being used (counting from the far left). This would be similar to an address 198.32.0.0. with a mask of 255.255.0.0.

A network is called a *supernet* when the prefix boundary contains fewer bits than the network's natural mask. For example, a class C network 192.56.10.0 has a natural mask of 255.255.255.0. The representation 192.56.0.0/16 has a shorter mask than the natural mask (16 is less than 24), so it is a supernet.

Figure 5 shows how CIDR can reduce the number of entries globally in Internet routing tables. A service provider has a group of customers with class C addresses that begin with 192.56. Despite this relationship, the service provider announces each of the networks individually into the global Internet routing mesh.

Figure 5: Routing With and Without CIDR



Adding and Deleting Addresses

This section provides information about adding or deleting IP addresses.

Multinetting is adding more than one IP address to an IP interface—that is, a primary address and one or more secondary addresses.

To make an interface unnumbered, see *Setting Up an Unnumbered Interface* on page 39.

Adding a Primary Address

- The primary address must be the first address added to the interface.
- Adding a new primary address overwrites the existing primary address.
- You can change a secondary address to be the primary address on an interface only via SNMP.
- An unnumbered address is always the primary address; adding an unnumbered address, therefore, overwrites any other numbered address.

Deleting a Primary Address

- You must always remove the primary address from an interface last.
- You cannot delete the primary address if the interface still has assigned secondary addresses.

Adding a Secondary (Multinet) Address

- You cannot add a secondary address until you add the primary address.
- You cannot add a secondary address to bridged Ethernet interfaces.
- You cannot change a primary address to a secondary address.
- An interface can have multiple secondary addresses.

Deleting a Secondary Address

- You must delete secondary addresses before deleting the primary address.

ip address Command

Use the following command to add addresses to or delete addresses from an interface:

ip address

- Use to add a primary address or to add secondary addresses to an interface.
- To add multiple addresses to a single IP interface, use the **secondary** keyword. (Remember, if you add an address using the **ip address** command and do not include the **secondary** keyword, the new address becomes the primary address.)
- You can specify the subnetwork mask value in either dotted decimal or prefix length notation.
- Example—Adds a primary address (192.168.2.77) and two secondary addresses (172.31.7.22 and 10.8.7.22); the Fast Ethernet interface now has addresses in three networks.

  host1(config)#**interface fastEthernet 0/0**
  host1(config-if)#**ip address 192.168.2.77 255.255.255.0**
  host1(config-if)#**ip address 172.31.7.22 255.255.255.0 secondary**
  host1(config-if)#**ip address 10.8.7.22 255.255.255.0 secondary**

NOTE: You can use this command in Interface Configuration mode, Subinterface Configuration mode, or Profile Configuration mode.

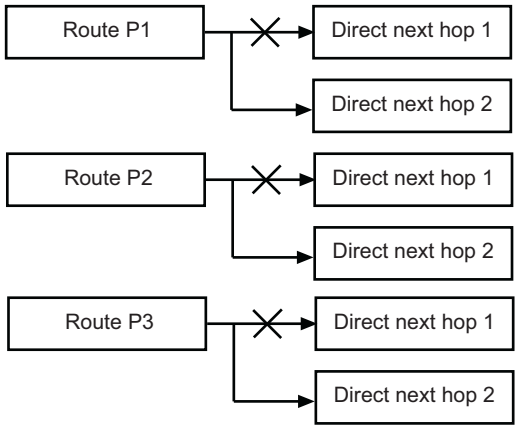
- Use the **no** version to remove an IP address. If you remove a primary IP address, IP processing is disabled on the interface.

Indirect Next-Hop Support

The router uses indirect next hops to promote faster network convergence (for example, in BGP networks) by decreasing the number of routing table changes required when a change in the network topology occurs.

Direct next-hops point routes in the routing table toward individual, direct next-hop connections. (See Figure 6.)

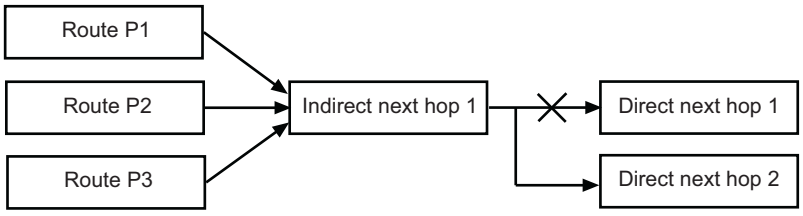
Figure 6: Direct Next Hops



Indirect next hops enable multiple routes in the routing table to point to a single next hop, thereby accelerating convergence. (See Figure 7.)

NOTE: Indirect next hops are not limited to any number of levels. In other words, an indirect next hop can point to a direct next hop or another indirect next hop.

Figure 7: Indirect Next Hops



By using indirect next hops, if a topology change occurs in the network, only the indirect next hop is modified in the routing table, decreasing the number of state changes required to achieve convergence.

Before You Configure IP

Before you configure IP, created lower-layer interfaces over which IP traffic flows.

For example, to configure an ATM interface:

```
host1(config)#interface atm 1/0
host1(config-if)#atm sonet stm-1
host1(config-if)#no loopback
host1(config-if)#atm clock internal chassis
host1(config-if)#interface atm 1/0.10
host1(config-if)#atm pvc 10 0 20 aal5snap
```

Refer to the appropriate chapters for information about configuring a specific type of interface.

NOTE: If you choose to configure VRRP, we recommend that you complete all IP address configurations before you configure VRRP. See *JUNOSe IP Services Configuration Guide, Chapter 14, Configuring VRRP*, for additional information.

Creating a Profile

You can configure an IP interface dynamically by creating a profile. A profile is a set of characteristics that acts as a pattern that can be dynamically assigned to an IP interface. You can manage a large number of IP interfaces efficiently by creating a profile with a specific set of characteristics. In addition, you can create a profile to assign an IP interface to a virtual router.

A profile can contain one or more of the following characteristics:

- access-route—Enables the creation of host access routes on an interface
- address—Configures an IP address on an interface
- auto-configure—Configures the interface for auto-configure mode
- auto-detect—Configures the interface for auto-detect mode
- directed-broadcast—Enables directed broadcast forwarding
- filter-options-all—Enables filtering of packets with IP options on an interface
- igmp—Configures an IGMP interface
- ignore-df-bit—Specifies that the don't-fragment bit is ignored
- inactivity-timer—Configures inactivity time for IP interfaces
- inspection—Associates an inspection list to the interface for firewalling
- mtu—Configures the maximum transmission unit for a network

- nat—Configures the interface as inside or outside for Network Address Translation (NAT)
- policy—Assigns a policy to the ingress or egress of an interface
- redirects—Enables transmission of ICMP redirect messages
- route-maps—Configures the interface for route-map processing
- source address validation—Verifies that a packet has been sent from a valid source address
- tcp adjust-mss—Adjusts maximum packet sizes on TCP connections when path MTU detection is not sufficient
- unnumbered—Configures IP on this interface without a specific address
- virtual-router—Specifies a virtual router to which interfaces created by this profile will be attached

Use the **profile** command from Global Configuration mode to create or edit a profile. See *JUNOSe Link Layer Configuration Guide, Chapter 12, Configuring Dynamic Interfaces* for information about creating profiles and on other characteristics that can be applied to the profile.

```
host1(config)#profile acton
host1(config-profile)#ip virtual-router warf
host1(config-profile)#ip unnumbered atm 3/0
```

ip access-routes

- Use to enable an access route in a profile.
- Example

```
host1(config)#profile foo
host1(config-profile)#ip access-routes
```

- Use the **no** version to remove the access route.

ip address

- Use to assign an IP address to a profile.
- You must first specify the layer 2 encapsulation before you can set the IP address for serial interfaces.
- Example

```
host1(config-if)#ip address 192.56.32.2 255.255.255.0
```

- Use the **no** version to remove the IP address assigned to the profile.

ip directed-broadcast

- Use to enable a directed broadcast address in a profile.
- Example

  host1(config-if)#**ip directed-broadcast**

- Use the **no** version to remove the directed broadcast address from the profile.

ip inspection

- Use to associate an inspection list to the inbound or outbound side of the IP interface.
- Example

  host1(config-profile)#**ip inspection list1**

- Use the **no** version to remove the inspection list association to this interface.

ip mtu

- Use to assign the MTU size sent on an IP interface.
- Example

  host1(config-if)#**ip mtu 5000**

- Use the **no** version to remove the assignment from the profile.

ip redirects

- Use to enable the sending of redirect messages if the software is forced to resend a packet through the same interface on which it was received.
- Example

  host1(config-if)#**ip redirects**

- Use the **no** version to remove the assignment from the profile.

ip tcp adjust-mss

- Use to modify the maximum segment size (MSS) for TCP SYN packets traveling through the interface. The router compares the MSS value of incoming or outgoing packets against the MSS adjustment value. For any packet that contains an MSS value larger than the MSS adjustment value, the router replaces the MSS option with the configured adjustment value. If the packet does not contain an MSS value, the router assumes a value of 536 for the packet MSS on which to base the comparison.

NOTE: The purpose behind using MSS is to alleviate problems with Path MTU Discovery (PMTUD) and resulting "black hole" detection issues. (See RFC 2923, "TCP Problems with Path MTU Discovery," for additional information about the "black hole" scenario.)

- Example

  host1(config-if)#**ip tcp adjust-mss 5000**

- Use the **no** version to remove the MSS assignment from the profile.

ip unnumbered

- Use to specify the numbered interface with which dynamic unnumbered interfaces created with the profile are associated.
- You can specify an unnumbered interface using RADIUS instead of using the **ip unnumbered** command in a profile.
- Example

  host1(config-profile)#**ip unnumbered fastEthernet 0/0**

- Use the **no** version to remove the assignment from the profile.

ip virtual-router

- Use to assign a virtual router to a profile.
- You can configure a virtual router using RADIUS instead of adding one to the profile by using the **ip virtual-router** command.
- Example

  host1(config-profile)#**ip virtual-router VR1**

- Use the **no** version to remove the virtual router assignment.

profile

- Use to create a profile.
- You specify a profile name with up to 80 characters.
- Example

  host1(config)#**profile foo**

- Use the **no** version to remove a profile.

Assigning a Profile

To assign a profile to an interface, use the **profile** command from Interface mode.

profile

- Use to assign a profile to a PPP interface. The profile configuration is used to dynamically create an upper IP interface.
- Example

  host1(config-if)#**interface serial 2/1**
  host1(config-if)#**encapsulation ppp**
  host1(config-if)#**profile acton**

- Use the **no** version to remove the assignment from the interface.

Address Resolution Protocol

Sending IP packets on a multiaccess network requires mapping from an IP address to a MAC address (the physical or hardware address).

In an Ethernet environment, Address Resolution Protocol (ARP) is used to map a MAC address to an IP address. ARP dynamically binds the IP address (the logical address) to the correct MAC address. Before IP unicast packets can be sent, ARP discovers the MAC address used by the Ethernet interface where the IP address is configured.

Hosts that use ARP maintain a cache of discovered Internet-to-Ethernet address mappings to minimize the number of ARP broadcast messages. To keep the cache from growing too large, an entry is removed if it is not used within a certain period of time. Before sending a packet, the host searches its cache for Internet-to-Ethernet address mapping. If the mapping is not found, the host sends an ARP request.

NOTE: For information about MAC address validation, see *MAC Address Validation* on page 22.

How ARP Works

Figure 8 and Figure 9 show how ARP works where host 1 sends an IP packet to host 2 on a different subnet. To complete this transmission, host 1 needs the MAC address of router 1, to be used as the forwarding gateway.

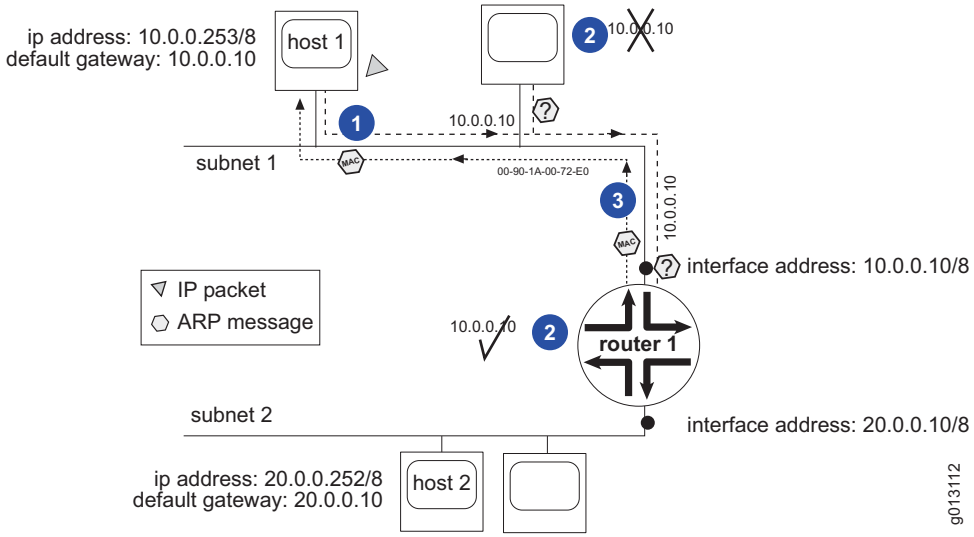
A typical scenario is:

1. Host 1 broadcasts an ARP request to all devices on subnet 1, composed by a query with the IP address of router 1. The IP address of router 1 is needed because host 2 is on a different subnet.

2. All devices on subnet 1 compare their IP address with the enclosed IP address sent by host 1.

3. Having the matching IP address, router 1 sends an ARP response, which includes its MAC address, to host 1.

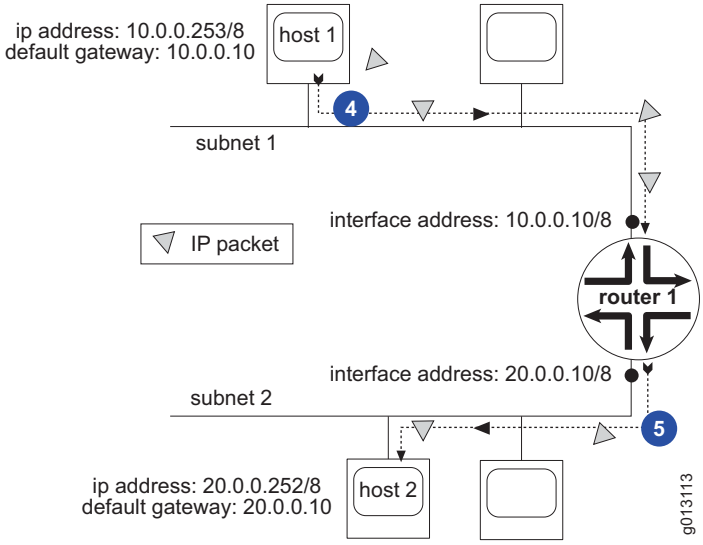
Figure 8: How ARP Works, Steps 1, 2, and 3



4. Host 1 transmits the IP packet to layer 3 DA (host 2) using router 1's MAC address.

5. Router 1 forwards IP packet to host 2. Router 1 might send an ARP request to identify the MAC of host 2. (See Figure 9.)

Figure 9: How ARP Works, Steps 4 and 5



ARP forces all receiving hosts to compare their IP addresses with the IP address of the ARP request. So if host 1 sends another IP packet to host 2, host 1 searches its ARP table for the router 1 MAC address.

If the default router/gateway becomes unavailable, then all the routing/packet forwarding to remote destinations ceases. Usually, manual intervention is required to restore connectivity, even though alternative paths may be available. Alternatively, Virtual Router Redundancy Protocol (VRRP) may be used to prevent loss of connectivity. See *JUNOSe IP Services Configuration Guide, Chapter 14, Configuring VRRP*.

arp

- Use to add a static (permanent) entry in the ARP cache.
- To add a static entry in the ARP cache, specify the *ipAddress*, *interfaceType* and *interfaceSpecifier (*as indicated in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide)*, and an optional MAC address
- You can issue this command only for Fast Ethernet interfaces, Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, and bridged Ethernet interfaces configured over ATM 1483.
- Example

  host1(config)#**arp 192.56.20.1 gig 2/0 0090.1a00.0170**

- Use the **no** version to remove an entry from the ARP cache.

arp timeout

- Use to specify how long an entry remains in the ARP cache.
- You can issue this command only for Fast Ethernet interfaces, Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, and bridged Ethernet interfaces configured over ATM 1483.
- The default value is 21,600 seconds (6 hours). Use the **show config** command to display the current value.
- If you specify a timeout of 0 seconds, entries are never cleared from the ARP cache.
- Example

  host1(config-if)#**arp timeout 8000**

- Use the **no** version to restore the default value.

clear arp

- Use to clear dynamic entries from the ARP cache.
- To clear a particular entry, specify all of the following:
  - *ipAddress*—IP address in four-part dotted-decimal format corresponding to the local data link address
  - *interfaceType*—Interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*
- To clear all dynamic ARP entries, specify an asterisk (*****).
- Example

  host1#**clear arp**

- There is no **no** version.

ip proxy-arp

- Use to enable proxy ARP on an Ethernet or bridge1483 interface.
- Proxy ARP is enabled by default.
- Example

  host1(config-if)#**ip proxy-arp unrestricted**

- Use the **no** version to disable proxy ARP on the interface.

MAC Address Validation

MAC address validation is a verification process performed on each incoming packet to prevent spoofing on IP Ethernet-based interfaces, including bridged Ethernet interfaces. When an incoming packet arrives on a layer 2 interface, the validation table is used to compare the packet's source IP address with its MAC address. If the MAC address and IP address match, the packet is forwarded; if it does not match, the packet is dropped.

NOTE: MAC address validation for bridged Ethernet interfaces is supported only on OC12a line modules.

MAC address validation on the E-series router can be accomplished in two ways:

- You can statically configure it on a physical interface via the **arp validate** command
- You can enable DHCP to perform the function independently and dynamically. See *DHCP* in *JUNOSe Link Layer Configuration Guide, Chapter 8, Configuring Bridged IP*.

The **arp validate** command adds the IP-MAC address pair to the validation table maintained on the physical interface.

If the validation is added statically via the CLI, the IP address–MAC address pairs are stored in NVS. The entries are used for MAC validation only if MAC validation is enabled on the interface via the **ip mac-validate** command.

CAUTION: When you configure an interface using the **arp validate** command, you cannot overwrite the ARP values that were added by DHCP.

You can enable or disable MAC address validation on a per interface basis by issuing the **ip mac-validate** command. See *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces* or *JUNOSe Link Layer Configuration Guide, Chapter 9, Configuring Bridged Ethernet* for information.

A dynamic IP subscriber interface inherits the MAC address validation state (enabled or disabled) configured for its parent static primary IP interface. See *Inheritance of MAC Address Validation State for Dynamic Subscriber Interfaces* in *JUNOSe Broadband Access Configuration Guide, Chapter 25, Configuring Subscriber Interfaces* for information.

arp validate

- Use to add IP address–MAC address validation pairs. When validation is enabled, all packets with the source IP address received on this IP interface are validated against the IP-MAC entries.
- To add a validation pair, specify one of the following:
  - *ipAddress* and *macAddress* of the interface
  - *ipAddress*, *interfaceType* and *interfaceSpecifier (*as indicated in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide)*, and an optional MAC address
- You can issue this command only for an IP Ethernet-based interface.
- For subscriber interface configurations, the IP address–MAC address pair must have a matching source prefix that already exists on the subscriber interface. If the matching source prefix does not exist, the IP–MAC address pair is rejected. See *JUNOSe Broadband Access Configuration Guide, Chapter 25, Configuring Subscriber Interfaces* for information about using subscriber interfaces.
- Example 1—Packets originating from host 192.56.20.1 and validated at Gigabit Ethernet interface with the MAC address 0090.1a00.0170

  host1(config)#**arp 192.56.20.1 gig 2/0 0090.1a00.0170 validate**

- Example 2—Subscriber interface MAC address validation enabled

  host1(config)#**arp 192.168.32.0 ip subsc1 000.0001.8100**

- Use the **no** version to remove an entry from the ARP cache.

Broadcast Addressing

A broadcast is a data packet destined for all hosts on a particular physical network. Network hosts recognize broadcasts by special addresses.

The router supports the following kinds of broadcast types:

- Limited broadcast—A packet is sent to a specific network or series of networks. A limited broadcast address includes the network or subnet fields. In a limited broadcast packet destined for a local network, the network identifier portion and host identifier portion of the destination address is either all ones (255.255.255.255) or all zeros (0.0.0.0).
- Flooded broadcast—A packet is sent to every network.
- Directed broadcast—A packet is sent to a specific destination address where only the host portion of the IP address is either all ones or all zeros (such as 192.20.255.255 or 190.20.0.0).

Several early IP implementations do not use the current broadcast address standard. Instead, they use the old standard, which calls for all zeros instead of all ones to indicate broadcast addresses. Many of these implementations do not recognize a broadcast address of all ones and fail to respond to the broadcast correctly. Others forward broadcasts of all ones, which causes a serious network overload known as a *broadcast storm*. Implementations that exhibit these problems include systems based on versions of BSD UNIX before version 4.3.

Routers provide some protection from broadcast storms by limiting their extent to the local cable. Bridges (including intelligent bridges), because they are layer 2 devices, forward broadcasts to all network segments, thus propagating all broadcast storms.

The best solution to the broadcast storm is to use a single broadcast address scheme on a network. Most IP implementations allow the network manager to set the address to be used as the broadcast address. Many implementations of IP, including the one on your router, can accept and interpret all possible forms of broadcast addresses.

Broadcast Tasks

You can use two broadcast-related IP commands to perform broadcast-related tasks.

ip broadcast-address

- Use to broadcast to all addresses in the host portion of an IP address.
- You specify an IP address to set the broadcast address.
- Example

  host1(config-if)#**ip broadcast-address 255.255.255.255**

- Use the **no** version to restore the default IP broadcast address.

ip directed-broadcast

- Use to enable translation of directed broadcasts to physical broadcasts.
- Example

  host1(config-if)#**ip directed-broadcast**

- Use the **no** version to disable the function.

Fragmentation

Fragmentation is the process of segmenting a large IP datagram into several smaller pieces. Fragmentation is required when IP must transmit a large packet through a network that transmits smaller packets, or when the MTU size of the other network is smaller.

By default, the router does not fragment the packet if the don't-fragment bit (DF bit) is set in the IP header. You can specify that the router not consider the DF bit before determining whether to fragment a packet.

NOTE: Lower-layer protocols can also set the MTU value. If MTU values set in lower layers differ from the one set at the IP layer, the router always uses the MTU lower-layer value.

ip ignore-df-bit

- Use to force the router to ignore the DF bit if it is set in the IP packet header for packets on an interface.
- Example

  host1(config-if)#**ip ignore-df-bit**

- Use the **no** version to restore the default behavior, which is to consider the DF bit before fragmentation.

ip mtu

- Use to set the MTU size of IP packets sent on an interface.
- The range is 128–10240.
- Do not configure both MLPPP fragmentation (with the **ppp fragmentation** command) and IP fragmentation of L2TP packets (with the **ip mtu** command) on the same interface. Instead, you must choose only one of the fragmentation configurations by setting it to the necessary value and set the other fragmentation configuration to the maximum allowable value.
- Example

  host1(config-if)#**ip mtu 1000**

- Use the **no** version to restore the default MTU size.

IP Routing

The Internet is a large collection of hosts that communicate with each other and use routers as intermediate packet switches.

Routers forward a packet through the interconnected system of networks and routers until the packet reaches a router that is attached to the same network as the destination host. The router delivers the packet to the specified host on its local network.

Routing Information Tables

A router makes forwarding decisions based on the information that is contained in its routing table. Routers announce and receive route information to and from other routers. They build tables of routes based on the collected information about all the best paths to all the destinations they know how to reach.

Each configured protocol has one or more local routing tables, sometimes referred to as a routing information base (RIB). This table is a database local to the protocol that contains all the routes known by that protocol to the prefixes in the table. For example, OSPF might have four different routes to 10.23.40.5.32. Only one of these routes is the best route to that prefix known to OSPF, but all four routes are in the OSPF local routing table.

The global routing table is a database maintained by IP on the SRP module. It contains at most one route per protocol to each prefix in the table. Each of these routes is the best route known by a given protocol to get to that prefix. The IP routing table does not, for example, have two OSPF routes to 10.5.11.0/24; it will have only one (if any) OSPF route to that prefix. It might also have a BGP route to the prefix, and a RIP route to the prefix, but no more than one route to a prefix per protocol.

IP compares the administrative distances for the routes to each prefix and selects the overall best route regardless of protocol. The best route to 10.5.11.0/24 might be via IS-IS. The best route to 192.168.0.0/16 might be via EBGP, and so on. These selected overall best routes to each prefix are used to create the forwarding table. The forwarding table is pushed to each line module. The line modules use their local instance of the forwarding table to forward the packets that they receive. When the global IP routing table is updated, so are the forwarding tables on the line modules.

Figure 10 illustrates a very simple network composed of three networks and two routers. The hosts that are attached to each network are not shown, because each router makes its forwarding decisions based on the network number and not on the address of each individual host. The router uses ARP to find the physical address that corresponds to the Internet address for any host or router on networks directly connected to it.

Figure 10: Routers in a Small Network

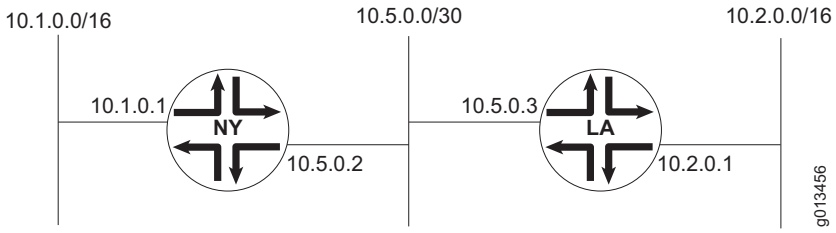


Table 4 and Table 5 represent information from the routing tables for routers NY and LA. Each routing table contains one entry for each route for each protocol or route type. Each routing table entry includes the following information:

- The destination IP network address.
- The IP address of the next-hop router.
- The type of network, such as static, directly connected, or the particular protocol.
- An administrative distance that is used to select the least-cost route among multiple routes to the same destination network. The least-cost (best) route is placed in the forwarding table. The administrative distance is not included in the forwarding table.
- A metric that is used by protocols to which the route is redistributed to select the least-cost route among multiple routes to the same destination network. The metric is not used to determine the best route to be placed in the forwarding table. The metric is also not listed in the forwarding table.

Table 4: Routing Table for Router NY

Destination Network	Next-Hop Router	Route Type	Administrative Distance	Metric
10.1.0.0/16	10.1.0.1	connected	0	0
10.2.0.0/16	10.5.0.3	OSPF	110	10
10.2.0.0/16	10.5.0.3	IS-IS	115	10
10.2.0.0/16	10.5.0.3	EBGP	20	15
10.2.0.0/16	10.5.0.3	RIP	120	5
10.5.0.0/30	10.5.0.2	connected	0	0

Table 5: Routing Table for Router LA

Destination Network	Next-Hop Router	Route Type	Administrative Distance	Metric
10.1.0.0/16	10.5.0.2	static	1	0
10.1.0.0/16	10.5.0.2	OSPF	110	10
10.1.0.0/16	10.5.0.2	RIP	120	4
10.2.0.0/16	10.2.0.1	connected	0	0
10.5.0.0/30	10.5.0.3	connected	0	0

Setting the Administrative Distance for a Route

The administrative distance is an integer that is associated with each route known to a router. The distance represents how reliable the source of the route is considered to be. A lower value is preferred over a higher value. An administrative distance of 255 indicates no confidence in the source; routes with this distance are not installed in the routing table.

Table 6 lists the default distance for each type of source from which a route can be learned.

Table 6: Default Administrative Distances for Route Sources

Route Source	Default Distance
Connected interface	0
Static route	1
Internal access route	2
Access route	3
External BGP	20
OSPF	110
IS-IS	115
RIP	120
Internal BGP	200
Unknown	255

If the IP routing table contains several routes to the same prefix—for example, an OSPF route and a RIP route—the route with the lowest administrative distance is used for forwarding.

To set the administrative distance for BGP routes, see *Setting the Administrative Distance for a Route* in *JUNOSe BGP and MPLS Configuration Guide, Chapter 1, Configuring BGP Routing*.

To set the administrative distance for RIP, IS-IS, and OSPF, use the following **distance** commands in Router Configuration mode.

distance

- Use to set an administrative distance for RIP or OSPF routes in the range 0–255.
- For RIP routes, the default value is 120.
- For OSPF routes, the default value is 110.
- Example

  host1(config)#**router rip**
  host1(config-router)#**distance 100**

- Use the **no** version to restore the default value.

distance ip

- Use to set the administrative distance for IS-IS routes in the range 1–255.
- Example

  host1(config)#**router isis**
  host1(config-router)#**distance 80 ip**

- Use the **no** version to restore the default value of 115.

Setting the Metric for a Route

For information about how to set a metric for a route, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy* as well as the individual routing protocol chapters in the *JUNOSe BGP and MPLS Configuration Guide*, and in this guide.

Routing Operations

Routers keep track of next-hop information that enables a data packet to reach its destination through the network. A router that does not have a direct physical connection to the destination checks its routing table and forwards packets to another next-hop router that is closer to that destination. This process continues until the packet reaches its final destination.

Identifying a Router Within an Autonomous System

The router ID is commonly one of the router's defined IP addresses. Although the router ID is, by convention, formatted as an IP address, it is not required to be a configured address of the router. If you do not use the **ip router-id** command to assign a router ID, the router uses one of its configured IP addresses as the router ID.

ip router-id

- Use to assign a router ID—a unique identifier that IP routing protocols use to identify the router within an autonomous system.
- Example

  host1(config)#**ip router-id 192.32.15.23**

- Use the **no** version to remove the router ID assignment.

Establishing a Static Route

You can set a destination to receive and send traffic by a specific route through the network.

ip route

- Use to establish a static route.
- Example

  host1(config)#**ip route 192.56.15.23 255.255.255.0 192.66.0.1**

- Use the **no** version to remove a static route from the routing table.

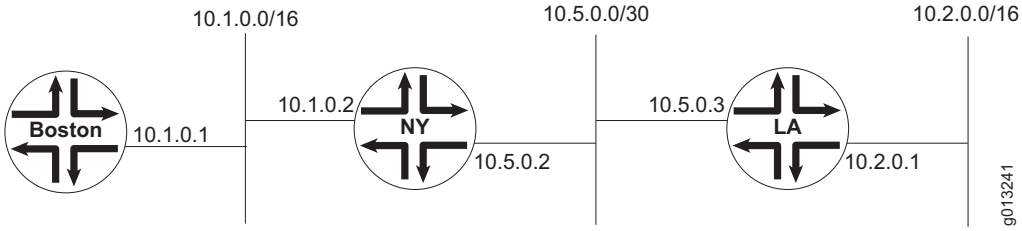
Configuring Static Routes with Indirect Next Hops

You can configure static routes where next hops are not on directly connected interfaces. Such a route is usable, and appears in the route table, only if another route in the route table can resolve the next hop.

The resolving route can be either statically created or dynamically learned by a routing protocol (like RIP, BGP, OSPF, and so on).

NOTE: When configuring this type of static route, the route that resolves the next hop must have an administrative distance value that is better (lower) than the distance of the static route you want to resolve.

Figure 11: Static Routes with Indirect Next Hops



On the Boston router in the network shown in Figure 11:

1. Configure a static route to 10.2.0.0/16 with a next hop of 10.5.0.2 (which is not directly connected) and an administrative distance of 254 (which is worse [higher] than the default administrative distance for static routes [1]).

   host1(config)#**ip route 10.2.0.0 255.255.0.0 10.5.0.2 254**

2. Configure another static route that resolves 10.5.0.2 and uses the default administrative distance.

   host1(config)#**ip route 10.5.0.0 255.255.255.252 10.1.0.2** [**1**]

NOTE: The previous example shows the default administrative distance value, 1, to illustrate the difference between the two static route commands. However, you do not have to enter this value when issuing the command.

A static route to 10.2.0.0 is added to the route table with a next hop of 10.1.0.2 (on the directly connected Ethernet interface).

NOTE: A dynamically learned route can also resolve indirect next hops, as long as the administrative distance value of the learned route is better (lower) than the static route whose next hop is being resolved.

Verifying Next Hops for Static Routes

You can configure either Bidirectional Forwarding Detection (BFD) or Response Time Reporter (RTR) probes to further control when a static route is installed in the routing table. Using either BFD or RTR, static route installation is based on two factors: whether the next hop to the specified destination is resolved, and whether an IP service running above the static route application is currently available.

Next-hop verification is useful for static route configurations in which the layer 2 and layer 3 interfaces are up and the next hop to the specified destination is available, but the IP services that run above the static route are currently unavailable. When the upper-layer IP services are unavailable, the router does not install the static route in its routing table.

How BFD Next-Hop Verification Works

Static routes on E-series routers can use Bidirectional Forwarding Detection (BFD) to verify the availability of the next hop and obtain the state of the IP service. For additional information about BFD, see *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.

If you specify the **bfd-liveness-detection** keywords with a minimum receive interval, minimum transmit interval, or multiplier when you issue the **ip route** command to establish a static route, the router verifies the next-hop status and installs the static route in the routing table under the following conditions:

- You configure the static routes with the actual next hop address and not just interface details.
- The BFD protocol is operational on both ends of the verification.
- The next hop is adjacent to the router (that is, only one hop away).

NOTE: BFD next-hop verification does not currently function in a multi-hop configuration.

- The next hop to the specified IP destination address is resolved.

You can further control the installation of static routes by specifying the **last-resort** keyword, which is valid when you use the **bfd-liveness-detection** keywords in the **ip route** command. The **last-resort** keyword instructs the router to install the static route in the routing table even if the specified BFD operation is unreachable, provided that no other static route to the same network prefix is available.

The static route is removed from the routing table if the next hop is no longer reachable and reinstalled when the route becomes reachable again.

BFD Next Hop Verification Configuration Example

To enable BFD next hop verification between two adjacent peers, you configure each peer as follows:

1. Configure peer A with the next hop address of peer B along with the desired intervals and keyword options.

   host1(config)#**ip route 192.1.1.0 255.255.255.0 192.1.2.1 verify bfd-liveness-detection minimum-interval 500 multiplier 3 last-resort**

2. Configure peer B with the next hop address of peer A along with the desired intervals and keyword options.

   host1(config)#**ip route 192.1.2.1 255.255.255.0 192.1.1.0 verify bfd-liveness-detection minimum-interval 300 multiplier 3**

ip route verify bfd-liveness-detection

- Use to enable BFD on a static route.
- Use the **minimum-interval** keyword to specify a value in the range 100–65535 milliseconds. This keyword defines both the minimum receive interval and minimum transmit interval using the same value.
- Use the **minimum-receive-interval** keyword to specify a minimum receive interval value in the range 100–65535 milliseconds.
- Use the **minimum-transmit-interval** keyword to specify a minimum transmit interval value in the range 100–65535 milliseconds.
- Use the **multiplier** keyword to specify a multiplier number in the range 1–255.
- Optionally, you can include the **last-resort** keyword when you use the **verify bfd-liveness-detection** keywords to instruct the router to install the static route in the routing table even if the specified BFD operation is currently unreachable, provided that no other static route to the same network prefix is available.
- Change parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each BFD peer.
- Example 1—Next hop address and last resort

  host1(config)#**ip route 192.56.15.23 255.255.255.0 192.66.0.1 verify bfd-liveness-detection minimum-interval 800 multiplier 2 last-resort**

- Example 2—Next hop address and interface

  host1(config)#**ip route 192.56.15.24 255.255.255.0 192.66.0.2 fast 6/0 verify bfd-liveness-detection**

- Example 3—Next hop address with different receive and transmit intervals

  host1(config)#**ip route 192.56.15.23 255.255.255.0 192.66.0.1 verify bfd-liveness-detection minimum-receive-interval 800 minimum-transmit-interval 300 multiplier 2 last-resort**

- Use the **no** version to remove the static route from the routing table and thereby remove BFD from that static route.

How RTR Next-Hop Verification Works

Static routes on E-series routers can use Response Time Reporter (RTR) probes configured as echo (ping) types to verify the availability of the next hop and obtain the state of the IP service. For more information about using RTR, see *Response Time Reporter* on page 63.

If you specify the **verify rtr** keywords with an RTR operation number when you issue the **ip route** command to establish a static route, the router verifies the next-hop status and installs the static route in the routing table only if *both* of the following conditions are met:

- The next hop to the specified IP destination address is resolved.
- The specified RTR operation is currently reachable.

You can further control the installation of static routes by specifying the **last-resort** keyword, which is valid only when you use the **verify rtr** keywords in the **ip route** command. The **last-resort** keyword instructs the router to install the static route in the routing table even if the specified RTR operation is unreachable, provided that no other static route to the same network prefix is available.

Although the configuration example in the next section uses Fast Ethernet interfaces, E-series routers support next-hop verification on any type of lower-layer interface.

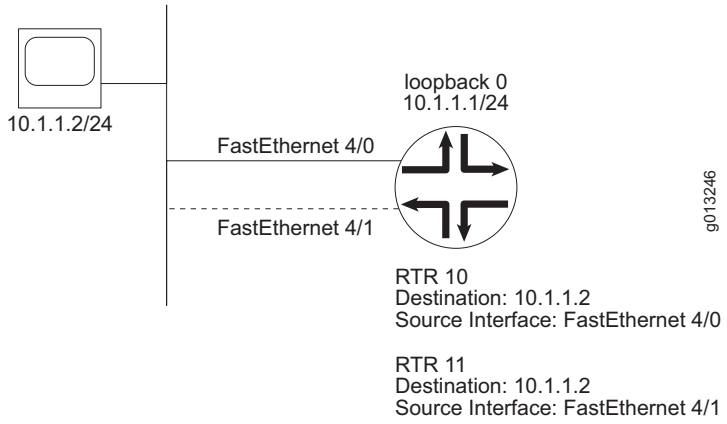
RTR Configuration Example

Figure 12 shows a sample configuration that illustrates the next-hop verification feature. In this example, two Fast Ethernet interfaces are configured between a remote system and an E-series router: Fast Ethernet interface 4/0 and Fast Ethernet interface 4/1. At any given time, only one of these interfaces forwards IP traffic, even though the associated layer 2 interfaces may be up concurrently.

On the E-series router, Fast Ethernet interfaces 4/0 and 4/1 are configured as unnumbered IP interfaces. In addition, each interface has an RTR probe configured as an echo type that sends requests over the interface to determine its availability. RTR 10 sends requests over Fast Ethernet interface 4/0, and RTR 11 sends requests over Fast Ethernet interface 4/1.

In this example, both RTR 10 and RTR 11 use the IP address of the remote system (10.1.1.2) as the target address. When you configure multiple RTR entries to use the same target address, you must set the **receive-interface** attribute to specify the interface on which the probe expects to receive responses. (See Step 4c in the next section, *Configuring RTR Next-Hop Verification*.) This action enables the router to map incoming responses to the proper RTR entry, even when multiple RTR entries have the same target address.

Figure 12: Sample Configuration for Next-Hop Verification



The **ip route** command is issued for each interface with the **verify rtr** and **last-resort** keywords to establish the necessary static routes. (See Steps 6 and 7 in the next section, *Configuring RTR Next-Hop Verification*.) This command causes the results described in Table 7, based on the status of the associated RTR operations.

Table 7: Next-Hop Verification Results for Sample Configuration

RTR 10 Status	RTR 11 Status	Results
Up	Up	The router installs an equal-cost multipath (ECMP) route to 10.1.1.2 in the routing table, using Fast Ethernet interfaces 4/0 and 4/1 as the next hops.
Up	Down	The router installs a route to 10.1.1.2, using Fast Ethernet interface 4/0 as the next hop.
Down	Up	The router installs a route to 10.1.1.2, using Fast Ethernet interface 4/1 as the next hop.
Down	Down	Although both RTR operations are down, the **last-resort** keyword instructs the router to install an ECMP route to 10.1.1.2, using Fast Ethernet interfaces 4/0 and 4/1 as the next hops. When all of the RTR operations associated with your static routes are down, you can control which route is installed in the routing table by including the **last-resort** keyword in the **ip route verify rtr** command only for the route that you want to install.

The next section, *Configuring RTR Next-Hop Verification*, provides instructions for configuring the example shown in Figure 12.

Configuring RTR Next-Hop Verification

To configure the next-hop verification example shown in Figure 12:

1. Configure a loopback interface, and assign an IP address and mask to the interface.

   ```
   host1(config)#interface loopback 0
   host1(config-if)#ip address 10.1.1.1 255.255.255.0
   host1(config-if)#exit
   ```

2. Configure Fast Ethernet port 4/0 with an unnumbered primary IP interface associated with the loopback interface configured in Step 1.

   ```
   host1(config)#interface fastEthernet 4/0
   host1(config-if)#ip unnumbered loopback 0
   host1(config-if)#exit
   ```

3. Repeat Step 2 for Fast Ethernet port 4/1.

   ```
   host1(config)#interface fastEthernet 4/1
   host1(config-if)#ip unnumbered loopback 0
   host1(config-if)#exit
   ```

4. Define probe RTR 10 for Fast Ethernet interface 4/0.

   a. Assign an operation number to the RTR probe, and access RTR Configuration mode. For information, see *Configuring the Probe Type* on page 64.

      ```
      host1(config)#rtr 10
      host1(config-rtr)#
      ```

   b. Configure the RTR probe as an echo type, and set the IP destination address and source interface.

      You must configure the RTR probe as an echo type to use next-hop verification. For information, see *Configuring the Probe Type* on page 64.

      ```
      host1(config-rtr)#type echo protocol ipIcmpEcho 10.1.1.2
      source fastEthernet 4/0
      ```

   c. Specify the interface on which the RTR probe expects to receive responses.

      You must set the **receive-interface** attribute when multiple RTR operations use the same target address. For information, see *Setting the Receiving Interface* on page 68.

      ```
      host1(config-rtr)#receive-interface fastEthernet 4/0
      ```

   d. (Optional) Configure optional probe characteristics, such as the frequency and samples-of-history kept. For information, see *Configuring Optional Characteristics* on page 65.

      ```
      host1(config-rtr)#frequency 1
      host1(config-rtr)#samples-of-history-kept 0
      ```

e. Exit RTR Configuration mode.

```
host1(config-rtr)#exit
```

f. Enable the probe to react to the test-failure event and the test-completion event.

You must configure both the test-failure and test-completion reaction conditions to use next-hop verification. For information, see *Setting Reaction Conditions* on page 69.

```
host1(config)#rtr reaction-configuration 10   test-failure 3
host1(config)#rtr reaction-configuration 10   test-completion
```

g. Schedule the probe operation. For information, see *Scheduling the Probe* on page 70.

```
host1(config)#rtr schedule 10 life 3
host1(config)#rtr schedule 10 restart-time 1
host1(config)#rtr schedule 10 start now
```

5. Repeat Step 4 to define RTR 11 for Fast Ethernet interface 4/1.

```
host1(config)#rtr 11
host1(config-rtr)#type echo protocol ipIcmpEcho 10.1.1.2
source fastEthernet 4/1
host1(config-rtr)#receive-interface fastEthernet 4/1
host1(config-rtr)#frequency 1
host1(config-rtr)#samples-of-history-kept 0
host1(config-rtr)#exit
host1(config)#rtr reaction-configuration 11 test-failure 3
host1(config)#rtr reaction-configuration 11 test-completion
host1(config)#rtr schedule 11 life 3
host1(config)#rtr schedule 11 restart-time 1
host1(config)#rtr schedule 11 start now
```

6. Establish a static route associated with RTR 10.

This command creates a static route and installs it in the routing table only if RTR 10 is currently reachable *or* if no other static route to IP destination address 10.1.1.2 is usable.

```
host1(config)#ip route 10.1.1.2 255.255.255.255 10.1.1.2 fastEthernet 4/0
verify rtr 10 last-resort
```

7. Establish a static route associated with RTR 11.

This command creates a static route and installs it in the routing table only if RTR 11 is currently reachable *or* if no other static route to IP destination address 10.1.1.2 is usable.

```
host1(config)#ip route 10.1.1.2 255.255.255.255 10.1.1.2 fastEthernet 4/1
verify rtr 11 last-resort
```

When both RTR 10 and RTR 11 are unreachable, you can control which static route is installed in the routing table by including the **last-resort** keyword in the **ip route verify rtr** command only for the route that you want to install.

NOTE: For detailed information about the commands for configuring RTR probes, see *Response Time Reporter* on page 63.

interface fastEthernet

- Use to select a Fast Ethernet (FE) interface on a line module or an SRP module.
- Example

  host1(config)#**interface fastEthernet 1/0**

- Use the **no** version to remove IP from an interface or subinterface. You must issue the **no** version from the highest level down; you cannot remove an interface or a subinterface if the one above it still exists.

NOTE: For more details on use of this command, see the syntax description in the *JUNOSe Command Reference Guide A to M*.

interface loopback

- Use to access and configure a loopback interface.
- You can use a loopback interface to provide a stable IP address that can minimize the impact if a physical interface goes down.
- Example

  host1(config)#**interface loopback 10**
  host1(config-if)#**ip address 100.20.32.1 255.255.255.0**

- Use the **no** version to delete the loopback interface.

ip address

- Use to set an IP address for an interface or a subinterface.
- Specify the layer 2 encapsulation before you set the IP address.
- Example

  host1(config-subif)#**ip address 192.0.2.50 255.255.255.0**

- Use the **no** version to remove the IP address or to disable IP processing on the interface.

ip route verify rtr

- Use to establish a static route and associate it with a configured RTR operation.
- Use the **verify rtr** keywords to instruct the router to install the static route in the routing table only if the next hop to the specified destination address is resolved and if the specified RTR operation is currently reachable. When you use the **verify rtr** keywords, you must also specify the number of the associated RTR operation.
- Optionally, you can include the **last-resort** keyword when you use the **verify rtr** keywords to instruct the router to install the static route in the routing table even if the specified RTR operation is currently unreachable, provided that no other static route to the same network prefix is available.
- Example

  host1(config)#**ip route 10.1.1.5 255.255.255.0 10.1.1.5 fastEthernet 1/0 verify rtr 5 last-resort**

- Use the **no** version to remove a static route from the routing table.

ip unnumbered

- Use to configure an unnumbered IP interface.
- This command enables IP processing on an interface without assigning an explicit IP address to the interface.
- You must specify an interface location, which is the identifier of another interface on which the router has an assigned IP address. This interface cannot be another unnumbered interface.
- Example 1

  host1(config-if)#**ip unnumbered fastEthernet 3/0**

- Example 2

  host1(config-if)#**ip unnumbered loopback 10**

- Use the **no** version to disable IP processing on the interface.

Setting Up Default Routes

A router examines its routing table to find a path for each packet. If the router cannot locate a route, it must discard the packet. You can set up a default route using the special address: 0.0.0.0. If the router cannot locate a path to a destination network and a default route is defined, the router forwards the packet to the default router. For example:

host1(config)#**ip route 0.0.0.0 0.0.0.0 192.56.21.3**

Default routes are typically used to reduce the size of the routing table. Routing is simplified because the router can test for a few local networks or use the default route. However, a disadvantage of default routes is the possible creation of multiple paths and routing loops.

Setting Up an Unnumbered Interface

An unnumbered interface does not have an IP address assigned to it. Unnumbered interfaces are often used in point-to-point connections where an IP address is not required.

ip unnumbered

- Use to set up an unnumbered interface.
- This command enables IP processing on an interface without assigning an explicit IP address to the interface.
- You supply an interface location, which is the type and number of another interface on which the router has an assigned IP address. This interface cannot be another unnumbered interface.
- Example

  host1(config-if)#**ip unnumbered fastEthernet 0/0**

- Use the **no** version to disable IP processing on an interface.

Adding a Host Route to a Peer on a PPP Interface

You can enable the ability to create host access routes on a PPP interface. This feature is useful in B-RAS applications.

ip access-routes

- Use to enable the ability to create host access routes on a PPP interface.
- Example

  host1(config-if)#**ip access-routes**

- Use the **no** version to disable this feature.

Enabling Source Address Validation

Source address validation verifies that a packet has been sent from a valid source address. When a packet arrives on an interface, the router performs a routing table lookup using the source address. The result from the routing table lookup is an interface to which packets destined for that address are routed. This interface must match the interface on which the packet arrived. If it does not match, the router drops the packet.

ip sa-validate

- Use to enable source address validation.
- Example

  host1(config-if)#**ip sa-validate**

- Use the **no** version to disable source address validation.

Enabling Source Address Validation Traps

The **ip sa-validate trap-enable** command enables the generation of traps for source address validation failure.

NOTE: To fully enable source address validation traps, you must also enable the IP trap category with the **snmp-server trap enable** command. See *Configuring Traps* on page 153 for more information.

ip sa-validate trap-enable

- Use to enable the generation of traps for source address validation failure on the router.
- You can specify a VRF context for which you want to enable trap validation for source address validation.
- Example

  host1(config)#**ip sa-validate trap-enable**

- Use the **no** version to disable the generation of source address validation failure traps on the router.

Defining TCP Maximum Segment Size

The **ip tcp adjust-mss** command enables you to modify the TCP maximum segment size (MSS) for TCP sessions.

When defined, the router modifies the maximum segment size (MSS) for TCP SYN packets traveling through the interface. The modification occurs only for packets that contain values smaller than the adjusted MSS value. When the packet does not contain an MSS field value, the router assumes an MSS value of 536 and makes any modifications accordingly.

NOTE: Implementation of the MSS value is identical for both ingress and egress interface traffic. That is, the router uses the same MSS value when adjusting inbound or outbound TCP traffic.

ip tcp adjust-mss

- Use to modify the maximum segment size (MSS) for TCP SYN packets traveling through the interface. The router compares the MSS value of incoming or outgoing packets against the adjusted MSS setting and replaces smaller values that it detects in any packets with the larger setting. If the packet does not contain an MSS value, the router assumes a value of 536 for the packet MSS on which to base the comparison.

NOTE: The purpose behind using MSS is to alleviate problems with Path Discovery (PMTUD) and resulting "black hole" detection issues. (See RFC 2923, "TCP Problems with Path MTU Discovery," for additional information about the "black hole" scenario.)

- Example

  host1(config-if)#**ip tcp adjust-mss 1000**

- Use the **no** version to remove the MSS assignment from the profile.

Setting MSS for TCP Connections

MSS is used by TCP to define the maximum amount of data that a TCP interface can accept in any single packet (or segment size). The MSS value is typically negotiated during connection establishment and is not renegotiated.

By default, the router uses an MSS value of 536 bytes and the advertised MSS is derived from the MTU of the transmitting interface. However, you can use the **tcp mss** command to set the MSS for TCP advertisements.

tcp mss

- Use to specify the MSS value for TCP to advertise.

NOTE: The MSS value is equal to the MTU value minus the IP and TCP headers, so the MSS value is generally 40 bytes less than the MTU.

- Use the *vrfName* variable to specify a VRF to which you want to assign the TCP MSS value.
- Example

  host1(config-if)#**tcp mss 1000**

- Use the **no** version to remove the MSS value so that the router uses the advertised MSS derived from the MTU of the output interface.

Configuring IP Path MTU Discovery

IP hosts transmit large amounts of data to other hosts using a series of IP datagrams. To best use resources, increase performance, and avoid difficult reassembly, hosts try to send datagrams that are as large as possible without requiring fragmentation anywhere along the path from the source to the destination. This datagram size is referred to as the *path MTU* (*PMTU*), and it is equal to the smallest MTU for each hop in the path.

Path MTU discovery is the process of discovering the PMTU value and using that value when transmitting TCP packets in datagrams.

Enabling PMTU Discovery

Use the **tcp path-mtu-discovery** command to enable PMTU discovery on the active virtual router.

tcp path-mtu-discovery

- Use to enable and configure path MTU discovery on the virtual router.
- Issue the command without any keywords to enable path MTU discovery.
- Issue the **age-timer** keyword to set the time (*minutes*) that TCP waits before attempting to increase the path MTU after receiving an ICMP Too Big message or after previously increasing the PMTU successfully (*minutes2*). The range of these two timers is 1–30 minutes. The timer defaults to 10 minutes.
- Issue the **age-timer infinite** keyword to disable PMTU aging functions.
- Example 1—Enables path MTU discovery

  host1:VR1(config)#**tcp path-mtu-discovery**

- Example 2—Sets path MTU discovery age timers differently

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 20 15**

- Example 3—Sets path MTU discovery age timers to the same value (5 minutes)

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 5**

- Example 4—Disables path MTU discovery age timers

  host1:VR1(config)#**tcp path-mtu-discovery age-timer infinite**

- Use the **no** version with a keyword to return the value to its default. Issue the **no** version without any keywords to disable path MTU discovery on the virtual router.

Limiting PMTU

You can limit calculated PMTU values within a range by using the **tcp path-mtu-discovery max-mtu** and **tcp path-mtu-discovery min-mtu** commands. When specifying PMTU limits, keep the following in mind:

- If a PMTU discovery value is lower than the configured minimum MTU setting, PMTU discovery is disabled for that connection.
- If a PMTU discovery value is larger than the configured maximum MTU setting, the configured maximum MTU setting is used.
- The maximum MTU setting must be greater than the minimum MTU setting.

tcp path-mtu-discovery max-mtu

- Use to limit the maximum MTU size used for the path MTU.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery max-mtu 512**

- Use the **no** version to remove any limitation so that the virtual router uses the path MTU discovery value.

tcp path-mtu-discovery min-mtu

- Use to specify the minimum MTU value used for the path MTU. If the discovered PMTU value is less than the minimum setting, path MTU discovery is disabled for this connection.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery min-mtu 255**

- Use the **no** version to remove any limitation so that the virtual router uses the discovered path MTU value.

Specifying Black Hole Thresholds

A black hole threshold is a limit to the number of times a virtual router can retransmit identical sequences of datagrams before the retransmissions are identified as a problem.

Some domains might be configured not to generate certain ICMP messages (like an ICMP destination unreachable message) or to filter all ICMP messages. Under these conditions, the source of oversized ICMP packets never learns that it is sending oversized packets. The device continues sending oversized packets that never get through. This behavior is often referred to as a *black hole*.

tcp path-mtu-discovery black-hole-detect-threshold

- Use to specify the minimum MTU value used for the path MTU. If the discovered PMTU value is less than the minimum setting, path MTU discovery is disabled for this connection.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery black-hole-detect-threshold 200**

- Use the **no** version to disable black hole threshold detection.

Shutting Down an IP Interface

You can disable an interface to the router at the IP level without removing it.

ip shutdown

- Use to shut down an IP interface.
- Example

  host1(config-if)#**ip shutdown**

- Use the **no** version to restart the interface.

Removing the IP Configuration

You can remove the IP configuration from an interface or subinterface.

no ip interface

- Use to remove the IP configuration from an interface or subinterface and disable IP processing on the interface.
- Example

  host1(config-if)#**no ip interface**

Clearing IP Routes

The router enables you to clear the specified routing entries from the routing table. You must specify the IP address prefix and the mask of the IP address prefix to clear specific routes. You can later reinstall the routes you have cleared.

clear ip routes

- Use to clear specified IP routes according to an IP prefix or a VPN routing and forwarding (VRF) table.
- Use an asterisk (*) to clear all dynamic routes from the routing table.
- Example

  host1#**clear ip routes ***

- There is no **no** version.

ip refresh-route

- Use to enable the owning protocols (BGP, IS-IS, OSPF) to reinstall routes removed from the IP routing table by the **clear ip routes** command.
- Example

  host1#**ip refresh-route**

- There is no **no** version.

Clearing IP Interfaces

The router enables you to clear the counters on one or more specified IP interfaces.

clear ip interface

- Use to clear a specified IP interface.
- Example

  host1#**clear ip interface pos 2/0**

- There is no **no** version.

Setting a Baseline

The router enables you to set a baseline for statistics on an IP interface.

baseline ip interface

- Use to set a baseline for a specified IP interface.
- Example

  host1#**baseline ip interface pos 2/0**

- There is no **no** version.

Disabling Forwarding of Packets

The router enables you to disable forwarding of packets on an SRP Ethernet interface.

ip disable-forwarding

- Use to disable forwarding of packets on the SRP Ethernet interface.
- The purpose of this command is to maintain router performance by maximizing the CPU time available for routing protocols. Although you can allow data forwarding on the SRP Ethernet interface, router performance will be affected.
- You see an error message if you try to set this command for interfaces other than the SRP Ethernet interface.
- Example

  host1(config-if)#**ip disable-forwarding**

- Use the **no** version to enable forwarding of packets on the interface.

Enabling Forwarding of Source-Routed Packets

IP packets are normally routed according to the destination address they contain based on the routing table at each hop through a path. The originator or source of the source-routed packets specifies the path (the series of hops) that the packets must traverse; the source makes the routing decisions. The source can specify either of the following types of source routing:

- *Strict-source* routing specifies every hop that the packet must traverse. The specified path consists of adjacent hops. The source generates an ICMP error if the exact path cannot be followed. For example, for a path going from source router A to router B to router C to router D, router A specifies a strict-source route as B, C, D.
- *Loose-source* routing specifies a set of hops that the packet must traverse, but not necessarily every hop in the path. That is, the specified hops do not have to be adjacent. For example, for a path going from source router A to router B to router C to router D, router A specifies a loose-source route as B, D or C, D, or B, C, D.

ip source-route

- Use to enable forwarding of source-routed packets in a VR or VRF.
- Forwarding is disabled by default in all VRs.
- Example

  host1(config)#**ip source-route**

- Use the **no** version to disable forwarding of source-routed packets on the VR or VRF.

Forcing an Interface to Appear Up

The router enables you to force an IP interface to appear as if it is up, regardless of the state of the lower layers.

ip alwaysup

- Use to force an IP interface to appear as up regardless of the state of lower layers.
- This command reduces route topology changes when the network attached to this link is single-homed.
- Example

  host1(config-if)#**ip alwaysup**

- Use the **no** version to make the interface appear in the current state.

Specifying a Debounce Time

You can set a debounce time that requires an IP interface to be in a given state—for example, up or down—for the specified time before the state is reported. This feature prevents a link that briefly goes up or down from causing an unnecessary topology change, for example by causing an interface down condition. However, note that increasing the debounce time increases the latency of updating the routing table to reflect an up or down change, and the latency of routing protocols propagating the state change.

ip debounce-time

- Use to set the interval in milliseconds for which an interface must maintain a given state before the state change is reported.
- Example

  host1(config)#**ip debounce-time 5000**

- Use the **no** version to remove the debounce time requirement.

Adding a Description

The router enables you to add a text description or an alias to a static IP interface or subinterface. Adding a description helps you identify the interface and keep track of interface connections. If no IP interface currently exists, then a static IP interface is automatically created on the current layer 2 interface and the description is applied to that static IP interface. You cannot assign a profile to a layer 2 interface that has a static interface configured above it.

ip description

- Use to assign a text description or an alias to an IP interface or subinterface.
- The description or alias can be a maximum of 256 characters.
- Use the **show ip interface** command to display the text description.

- Example 1

  host1(config-if)#**ip description canada01 ip interface**

- Example 2

  host1(config-subif)#**ip description montreal011 ip subinterface**

- Use the **no** version to remove the text description or alias.

Enabling Link Status Traps

The router enables you to enable link status traps on an interface.

snmp trap ip link-status

- Use to enable link status traps on an interface.
- Example

  host1(config-if)#**snmp trap ip link-status**

- Use the **no** version to disable link status traps on an interface.

Configuring the Speed

The router enables you to set the speed of an IP interface.

ip speed

- Use to set the speed of the interface in bits per second.
- By default, the speed is determined from a lower-layer interface.
- Example

  host1(config-if)#**ip speed 1000**

- Use the **no** version to set the speed to the default, 0.

Configuring Equal-Cost Multipath Load Sharing

Equal-cost multipath (ECMP) sets are formed when the router finds routing table entries for the same destination with equal cost. The router then balances traffic across these sets of equal-cost paths by using one of two ECMP modes—hashed (the default) or round-robin.

Hashed mode uses hashing of source and destination addresses to determine which of the available paths in the ECMP set to use. Hashed mode is the default ECMP mode of operation.

Defining Maximum Paths

You can add routing table entries manually (as static routes), or they are formed as routers discover their neighbors and exchange routing tables (via OSPF, BGP, and other routing protocols).

The **maximum paths** command controls the maximum number of parallel routes that the routing protocol (BGP, IS-IS, OSPF, or RIP) can support.

Round-Robin Mode

Round-robin mode distributes packets equally among the available paths in the ECMP set.

ip multipath round-robin

- Use to specify round-robin as the mode for ECMP load sharing on an interface.
- ECMP uses the round-robin mode when you have configured *all* interfaces in the set to round-robin. Otherwise, ECMP defaults to hashed mode because round-robin mode can cause reordering of packets. You must explicitly ensure that the possible reordering is acceptable on all the member interfaces by setting them to round-robin mode.
- If one of the ECMP next hops is an indirect next hop, ECMP uses hashed mode load balancing.
- Example

  host1(config)#**virtual-router router_0**
  host1:router_0(config)#**interface serial 4/0:1/22.22**
  host1:router_0(config-subif)#**ip multipath round-robin**
  host1:router_0(config-subif)#**exit**

- Use the **no** version to set the ECMP mode to the default, hashed.

maximum-paths

- Use to control the maximum number of parallel routes that the routing protocol supports.
- The maximum number of routes can be in the range 1–16 for BGP, IS-IS, OSPF, or RIP.
- Example

  host1(config-router)#**maximum-paths 2**

- Use the **no** version to restore the default value, 1 for BGP or 4 for IS-IS, OSPF, or RIP.

Fast Reroute Protection

If a link goes down, ECMP uses fast reroute protection to shift packet forwarding to use operational links, thereby decreasing packet loss. Fast reroute protection updates ECMP sets for the interface without having to wait for the route table update process. When the next route table update occurs, a new ECMP set can be added with fewer links or the route might point to a single next hop.

CAUTION: To provide ECMP fast reroute functionality in the event of an interface failure, the members of an equal cost multipath must be resolved to corresponding interfaces. If the member is an indirect next hop, the interface is obtained by using the forwarding equivalence class (FEC) to which the member points. This method of resolving members occurs only if the FEC, pointed to by the indirect next hop, is either an interface or a direct next hop.

An indirect next hop member is not resolved to an interface if it points to another indirect next hop or to an equal cost multipath. ECMP fast reroute functionality is not available if any interfaces that correspond to unresolved indirect next hop members go down.

If you modify an indirect next hop member to point to a different FEC (that is, a different interface, direct next hop, indirect next hop, or ECMP), the indirect next hop member is not resolved for the new changes.

Setting a TTL Value

You can use the **ip ttl** command to set the TTL (time-to-live) field in the IP header for all IP operations. The TTL specifies a hop count. This configured TTL value can be overridden by other commands that specify a TTL.

ip ttl

- Use to set a default value for the IP header TTL field for all IP operations.
- Example

  host1(config)#**ip ttl 255**

- Use the **no** version to restore the default value, 127.

Protecting Against TCP RST or SYN DoS Attacks

You can use the **tcp ack-rst-and-syn** command to help protect the router from denial of service (DoS) attacks.

Normally, when it receives an RST or SYN message, TCP attempts to shut down the TCP connection. This action is expected under normal conditions, but someone maliciously generating valid RST or SYN messages can cause problems for TCP and the network as a whole.

When you enable the **tcp ack-rst-and-syn** command, the router challenges any RST or SYN messages that it receives by sending an ACK message back to the expected source of the message. The source reacts in one of the following ways:

- If the source did send the RST or SYN message, it recognizes the ACK message to be spurious and resends another RST or SYN message. The second RST or SYN message causes the router to shut down the connection.
- If the source did not send the RST or SYN message, the source accepts the ACK message as part of an existing connection. As a result, the source does not send another RST or SYN message and the router does not shut down the connection.

NOTE: Enabling this command slightly modifies the way TCP processes RST or SYN messages to ensure that they are genuine.

tcp ack-rst-and-syn

- Use to help protect the router from TCP RST and SYN denial of service attacks.
- Example

  host1(config)#**tcp ack-rst-and-syn**

- Use the **no** version to disable this protection.

Preventing TCP PAWS Timestamp DoS Attacks

The TCP Protect Against Wrapped Sequence (PAWS) number option works by including the TCP timestamp option in all TCP headers to help validate the packet sequence number.

Normally, in PAWS packets that have the timestamps option enabled, hosts use an internal timer to compare the value of the timestamp associated with incoming segments against the last valid timestamp the host recorded. If the segment timestamp is larger than the value of the last valid timestamp, and the sequence number is less than the last acknowledgement sent, the host updates its internal timer with the new timestamp and passes the segment on for further processing.

If the host detects a segment timestamp that is smaller than the value of the last valid timestamp or the sequence number is greater than the last acknowledgement sent, the host rejects the segment.

A remote attacker can potentially determine the source and destination ports and IP addresses of both hosts that are engaged in an active connection. With this information, the attacker might be able to inject a specially crafted segment into the connection that contains a fabricated timestamp value. When the host receives this fabricated timestamp, it changes its internal timer value to match. If this timestamp value is larger than subsequent timestamp values from valid incoming segments, the host determines the incoming segments as being too old and discards them. The flow of data between hosts eventually stops, resulting in a denial of service condition.

Use the **tcp paws-disable** command to disable PAWS processing.

NOTE: Disabling PAWS does not disable other processing related to the TCP timestamp option. This means that even though you disable PAWS, a fabricated timestamp that already exists in the network can still pollute the database and result in a successful DoS attack. Enabling PAWS resets the saved timestamp state for all connections in the virtual router and stops any existing attack.

tcp paws-disable

- Use to disable the Protect Against Wrapped Sequence (PAWS) number option in TCP segments.
- You can specify a VRF context for which you want PAWS disabled.
- Example

  host1(config)#**tcp paws-disable**

- Use the **no** version to restore PAWS processing (the default mode).

Protecting Against TCP Out of Order DoS Attacks

You can use the group of **tcp resequence-buffers** commands to help protect the router from TCP out-of-order DoS attacks.

TCP guarantees that applications receive data in order. This means that TCP buffers any out-of-order packets it receives until ordered delivery can occur. To prevent buffers from consuming too many resources, TCP limits the amount of data it accepts to the number of data bytes that the receiver is willing to receive and buffer.

TCP does not take into account the buffering scheme that the receiver uses. If the receiver uses a fixed-size receive buffer (that is, buffering all packets) regardless of length, a packet that contains only one data byte might consume many data bytes of buffer space, but only one byte of TCP space.

Under these conditions, an attacker can send a large number of 1-byte packets to an E-series router in which each packet is buffered, consuming an entire packet buffer and eventually consuming a large amount of resources.

To defend against this sort of attack, you can set defaults and limits on the number of outstanding buffers on reordering queues. You can configure these defaults and limits on a per-router, per-virtual router, or per-connection basis.

Limiting Buffers per Router

The **tcp resequence-buffers global-maximum** command enables you to limit the number of outstanding buffers on the entire router.

tcp resequence-buffers global-maximum

- Use to specify a router-wide maximum number of buffers that resequencing queues can contain.
- Specify a value of zero (0) to turn off the limit.
- Example

  host1(config)#**tcp resequence-buffers global-maximum**

- Use the **no** version to revert the global maximum buffer value to its default, 1000 buffers.

Limiting Buffers per Virtual Router

The **tcp resequence-buffers default-vr-maximum** command and **tcp resequence-buffers vr-maximum** command enable you to limit the number of outstanding buffers on existing or newly established virtual routers.

tcp resequence-buffers default-vr-maximum

- Use to specify the default buffer limit assigned to all virtual routers when the virtual router is established.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers default-vr-maximum 200**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

tcp resequence-buffers vr-maximum

- Use to define the maximum number of buffers that the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers vr-maximum**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

Limiting Buffers per Connection

The **tcp resequence-buffers connection-maximum** command and **tcp resequence-buffers default-connection-maximum** command enable you to limit the number of outstanding buffers on existing or newly established connections.

tcp resequence-buffers connection-maximum

- Use to define the maximum number of buffers that connections on the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the connection maximum.
- Example

  host1(config)#**tcp resequence-buffers connection-maximum 50**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

tcp resequence-buffers default-connection-maximum

- Use to specify the default buffer limit assigned to all TCP connections on a virtual router unless a specific limit is set for the VR in which the connection is established.
- Specify a value of zero (0) buffers to turn off the default limit.
- Example

  host1(config)#**tcp resequence-buffers default-connection-maximum 100**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

Distributing Routing Table Updates to Line Modules

You can configure the forwarding table hold-down time allotted after a routing table change for the accumulation of additional updates and the subsequent distribution of the set of routing table changes to the line modules.

forwarding-table route-holddown

- Use to enhance SRP performance by increasing the hold-down time allotted for accumulating and distributing sets of routing table changes to the line modules.
- A higher timer value can enhance SRP performance, but it can also delay the implementation of routing table changes on the line modules. Be aware of the possible effect on network performance before you reconfigure the forwarding table hold-down timer.
- Setting the hold-down timer to zero (0) distributes an update after each change to the routing table, which can degrade SRP performance.
- Example

  host1(config)#**forwarding-table route-holddown 15**

- Use the **no** version to set the hold-down timer to the default value, 3 seconds.

IP Tunnel Routing Table

The IP tunnel routing tables include IPv4 routes that point only to tunnels, such as MPLS tunnels. The tunnel routing table is not used for forwarding. Instead, protocols resolve next hops by looking up the routes that point to tunnels. The routes in the tunnel routing table cannot be redistributed. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS* for more information.

Shared IP Interfaces

You can create multiple *shared* IP interfaces over the same layer 2 logical interface—for example, atm 5/3.101—enabling more than one IP interface to share the same logical resources. You can configure one or more shared IP interfaces. Data sent over shared interfaces uses the same layer 2 interface. You can configure shared interfaces as you would unshared IP interfaces. Each shared interface has its own statistics.

Some layer 2 interfaces require a primary IP interface to negotiate certain IP parameters—for example, IPCP for PPP, ARP for Ethernet, and Inverse ARP for Frame Relay. If you do not configure a primary IP interface in such cases, the layer 2 interface cannot become operationally up.

A primary IP interface is the default interface for receiving data that arrives on the layer 2 interface. If you configure shared IP interfaces for the same layer 2 interface as your primary IP interface, by default data received on the layer 2 interface is received on the virtual router corresponding to the primary IP interface. A primary IP interface and all of its shared IP interfaces have the same interface location. You can configure a shared IP interface to receive data on the same layer 2 interface as a primary IP interface. You can delete primary and shared IP interfaces independently of each other.

You can create a primary IP interface as you do any other IP interface, as shown in the following example:

```
host1(config)#virtual-router vr-a:vrf-2
host1:vr-a:vrf-2:(config)#interface atm 5/3.101
host1:vr-a:vrf-2:(config-if)#ip address 10.1.1.1 255.255.255.255
host1:vr-a:vrf-2:(config-if)#exit
```

You do not have to configure a primary IP interface if you do not need one as described above. In the absence of a primary interface, you can still configure shared IP interfaces; however, in this scenario, data received on the layer 2 interface is discarded.

You cannot create shared IP interfaces for the following kinds of interface:

- IP floating interfaces (IP interfaces that stack over MPLS stacked tunnels)
- Loopback interfaces
- Null interfaces

For information about configuring shared IP interfaces to receive data on the same layer 2 interface as a primary IP interface, see *JUNOSe Broadband Access Configuration Guide, Chapter 25, Configuring Subscriber Interfaces*.

Configuring Shared IP Interfaces

To share IP interfaces:

1. Create a layer 2 interface.

   host1(config)#**interface atm 5/3**
   host1(config-if)#**interface atm 5/3.101**

2. (Optional) Create a primary IP interface.

   host1(config-if)#**ip address 10.1.1.1 255.255.255.255**
   host1(config-if)#**exit**

3. Create the shared IP interface.

   host1(config)#**interface ip si0**

4. Associate the shared IP interface with the layer 2 interface by one of the following methods:

   - Statically

     host1(config-if)#**ip share-interface atm 5/3.101**

   - Dynamically

     host1:vr-a:vrf-1(config-if)#**ip share-nexthop 10.0.0.1**

5. To fully configure the shared interface, assign an address (or make the interface unnumbered).

   host1(config-if)#**ip address 2.2.2.2 255.0.0.0**

interface ip

- Use to create an IP interface for interface sharing.
- Use the specified name to refer to the shared IP interface; you cannot use the layer 2 interface to refer to them, because the shared interface can be moved.
- Example

  host1(config)#**interface ip si0**

- Use the **no** version to delete the IP interface.

ip share-interface

- Use to specify the layer 2 interface used by a shared IP interface. The command fails if the layer 2 interface does not yet exist. The command is not supported (that is, it fails) if you use an RSVP tunnel (for example, **tunnel mpls:1**) to identify the layer 2 interface.
- If you issue this command on a shared IP interface, you cannot issue the **ip share-nexthop** command for the interface.
- After creating the shared IP interface, you can configure it as you do any other IP interface.

- The shared interface is operationally up when the layer 2 interface is operationally up.
- You can create operational shared IP interfaces in the absence of a primary IP interface.
- Example

  host1(config-if)#**ip share-interface atm 5/3.101**

- Use the **no** version to remove the association between the layer 2 interface and the shared IP interface. You can delete shared and primary IP interfaces independently.

ip share-nexthop

- Use to specify that the shared IP interface dynamically tracks a next hop. If the next hop changes, the shared IP interface moves to the new layer 2 interface associated with the IP interface toward the new next hop.
- If you issue this command on a shared IP interface, you cannot issue the **ip share-interface** command for the interface.
- If you issue this command on a shared IP interface, the shared interface cannot dynamically track the next hop for the specified destination if the next-hop IP address is resolvable over MPLS.
- If you specify a virtual router, the command fails if the VR does not already exist. If you do not specify a VR, the current VR is assumed.
- After creating the shared IP interface, you can configure it as you do any other IP interface.
- The shared interface is operationally up when the layer 2 interface associated with the specified next hop is operationally up. However, if the layer 2 interfaced associated with the specified next hop is an MPLS next hop (for example, an RSVP or LDP tunnel), the shared interface remains operationally down.
- Use the **no** version to halt tracking of the next hop.

Moving IP Interfaces

You can move an IP shared interface from one layer 2 interface to another by issuing the **ip share-interface** command to specify a different layer 2 interface. Moving an IP interface does not affect interface statistics, packets forwarded through the interface, or policies attached to the IP interface.

Example The following commands create shared interface si0 on the layer 2 interface atm5/3.101:

```
host1(config)#virtual-router vr-a:vrf-1
host1:vr-a:vrf-1(config)#interface ip si0
host1:vr-a:vrf-1(config-if)#ip share-interface atm 5/3.101
host1:vr-a:vrf-1(config-if)#exit
```

The following commands move shared interface si0 to the layer 2 interface atm5/3.201:

```
host1:vr-a:vrf-1(config)#interface ip si0
host1:vr-a:vrf-1(config-if)#ip share-interface atm 5/3.201
```

IP Shared Interface Statistics

Each shared interface has its own statistics. Packets transmitted on a shared IP interface are always counted only in the shared IP interface.

Subscriber Interfaces

A subscriber interface is an extension of a shared IP interface. Shared IP interfaces are unidirectional—they can transmit but not receive traffic. In contrast, subscriber interfaces are bidirectional—they can both receive and transmit traffic.

For details about configuring and using subscriber interfaces, see *JUNOSe Broadband Access Configuration Guide, Chapter 25, Configuring Subscriber Interfaces*.

Internet Control Message Protocol

IP was not designed to provide reliable delivery service. The higher-layer protocols that operate as clients of IP implement their own reliability procedures if reliable communications are required.

Internet Control Message Protocol (ICMP) provides a mechanism that enables a router or destination host to report an error in data traffic processing to the original source of the packet. ICMP messages provide feedback about problems that occur in the communication environment.

ICMP messages are sent only when errors occur in either the processing of an unfragmented data packet or the first fragment of a fragmented data packet.

ICMP messages are encapsulated as part of the data portion of an IP data packet and are routed like any other IP data packets. Thus, there is no guarantee to the sender of an ICMP message that the message will be delivered to its destination.

The router supports ICMP redirects. When a packet enters an IP interface and exits the same interface, the router may send an ICMP message to the originator of the packet. This message notifies the originator that a better gateway exists to the assigned destination address.

With the **ip redirects** command (used in Interface Configuration mode) you can enable or disable ICMP redirects. This attribute is enabled by default. If it is enabled on the IP interface and if the internal ICMP redirect queue is not full, the router sends an ICMP redirect packet to the originator. When the originator receives the ICMP redirect notification, the originator determines whether to start using the better gateway.

ICMP Tasks

You can enable the following ICMP features:

- ICMP Router Discovery Protocol (IRDP)
- ICMP netmask reply
- Sending of IP redirects
- Generation of ICMP unreachable messages

ip irdp

- Use to enable IRDP processing on an interface.
- Example

  host1(config-if)#**ip irdp**

- Use the **no** version to disable the function.

ip mask-reply

- Use to enable ICMP netmask reply.
- Example

  host1(config-if)#**ip mask-reply**

- Use the **no** version to disable the function.

ip redirects

- Use to enable the sending of redirect messages if software is forced to resend a packet through the same interface on which it was received.
- Example

  host1(config-if)#**ip redirects**

- Use the **no** version to disable the sending of redirect messages.

ip unreachables

- Use to enable the generation of an ICMP unreachable message when a packet is received that the router cannot deliver.
- Example

  host1(config-if)#**ip unreachables**

- Use the **no** version to disable the function.

Specifying a Source Address for ICMP Messages

By default, ICMP uses the IP address of the outgoing interface as the source IP address for the ICMP message. However, you can use the **ip icmp update-source** command to instruct ICMP to use an already configured interface or a specified IP address, as the source address of the ICMP message.

For example, you can specify that ICMP use Fast Ethernet interface 1 in slot 0 as the source for any messages that it sends:

host1(config)#**ip icmp update-source fastEthernet 0/1**

You must use an already configured interface or an existing IP address when using the **ip icmp update-source** command. Also, you cannot specify a multicast address when using this command.

ip icmp update-source

- Use to define an update source address for all ICMP messages that the E-series router generates from the specified interface.
- Example

  host1(config)#**ip icmp update-source 192.35.42.1**

- Use the **no** version to disable the function.

Reachability Commands

Use the **ping** and **traceroute** commands to determine reachability of destinations in the network.

- Use the **ping** command to send an ICMP or ICMPv6 echo request packet. In the following example, the request packet is sent to IP address 192.35.42.1, with a packet count of 10 and a timeout value of 10 seconds:

  host1#**ping 192.35.42.1 10 timeout 10**

- Use the **traceroute** command to discover routes that router packets follow when traveling to their destination. In the following example, the trace destination IP address is 192.56.20.1, the maximum number of hops of the trace is 20, and the timeout value is 10 seconds:

  host1#**traceroute 192.56.20.1 20 timeout 10**

ping

- Use to send an ICMP or ICMPv6 echo request packet to the IP address that you specify.
- You can specify a VRF context.
- Use the **source interface** keywords to specify a source interface other than the one from which the probe originates.
- Use the **source address** keywords to specify a source IP address other than the one from which the probe originates.
- You can specify the following options:
  - *packetCount*—Number of packets to send to the destination IP address. If you specify a zero (0), echo requests packets are sent indefinitely.
  - **data-pattern**—Sets the type of bits contained in the packet to all ones, all zeros, a random mixture of ones and zeros, or a specific hexadecimal data pattern that can range from 0x0–0xFFFFFFFF. The default is all zeros.
  - **data-size**—Sets the number of bytes comprising the IP packet and reflected in the IP header in the range 0–64000; the default is 100 bytes.
  - **extended** header attributes—Set the following:
    - A value to be set in the type of service (ToS) byte, in the range 0–255, to support quality of service (QoS) offerings
    - Don't-fragment bit to prevent IP from fragmenting the packet if it is too long for the MTU of a given link; if the nonfragmented packet cannot be delivered, it is discarded.
    - Strict-source or loose-source routing, including the IP address of the hops the packets must traverse. For loose-source-route, you specify some or all of the hops, but they do not have to be adjacent. For strict-source-route, you must specify every adjacent hop through which the packet must traverse.
    - The IP addresses to be recorded for a specified number of routers that the packets traverse.
    - The time that a packet traverses a router to be recorded for a specified number of routers.
    - An interface type and specifier of a destination address on the router that is configured for external loopback; the command succeeds only if the specified interface is configured for external loopback.
    - The traffic class value to match in the Traffic Class field of each packet (IPv6 only)
    - The flow label value to match in the Flow Label field of each packet (IPv6 only)
  - **sweep-interval**—Specifies the change in the size of subsequent ping packets while sweeping across a range of sizes. For example, you can configure the sweep interval to sweep across the range of packets from 100 bytes to 1000 bytes in increments equal to the sweep interval. By default the router increments packets by one byte; for example, it sends 100, 101, 102, 103, ... 1000. If the sweep interval is 5, the router sends 100, 105, 110, 115, ... 1000.

- **sweep-sizes**—Enables you to vary the sizes of the echo packets being sent. This capability is useful for determining the minimum sizes of the MTUs configured on the nodes along the path to the destination address. Determining the minimum size reduces packet fragmentation, which contributes to performance problems. The default is not to sweep (all packets are the same size).
- **timeout**—Sets the number of seconds to wait for an ICMP echo reply packet before the connection attempt times out.
- **ttl**—Sets the time-to-live hop count in the range 1–255; the default is 32.

- The following characters can appear in the display after issuing the **ping** command:
  - !—Reply received
  - .—Timed out while waiting for a reply
  - ?—Unknown packet type
  - A—Address mask request message
  - a—Address mask reply message
  - D—Router discovery advertisement message
  - d—Router discovery request message
  - H—Host unreachable
  - I—Information request message
  - i—Information reply message
  - L—TTL expired message
  - M—Could not fragment, DF bit set
  - m—Parameter problem message
  - N—Network unreachable
  - P—Protocol unreachable
  - Q—Source quench
  - r—Redirect message
  - T—Timestamp request message
  - t —Timestamp reply message
  - U—Destination unreachable
- Example

  host1(config)#**interface serial 5/2:1/1**
  host1(config-if)#**ip address 172.16.1.1 255.255.255.0**
  host1(config-if)#**exit**
  host1#**ping 172.16.1.1 extended interface serial 5/2:1/1**

- There is no **no** version.

traceroute

- Use to discover the routes that router packets follow when traveling to their destination.
- You can specify:
  - A VRF context
  - Destination IP or IPv6 address
  - Source interface for each of the transmitted packets
  - Source address for each of the transmitted packets
  - Maximum number of hops of the trace and a timeout value
  - Size of the IP packets (not the ICMP payload) in the range 0–64000 bytes sent with the **traceroute** command. Including a size might help locate any MTU problems that exist between your router and a particular device.
  - Extended IP header attributes, including the ToS byte (IP only), whether to set the DF bit for the transmitted packets (IP only), the traffic class (IPv6 only), and flow label (IPv6 only).
- You can also force transmission of the packets on a specified interface regardless of what the IP address lookup indicates.
- Example

  host1#**traceroute 172.20.13.1 20 timeout 10**

- There is no **no** version.

Response Time Reporter

The Response Time Reporter (RTR) feature enables you to monitor network performance and resources by measuring response times and the availability of your network devices.

RTR configuration is associated with a specific virtual router, distinct from any other virtual router.

Configuration Tasks

To configure RTR:

1. Configure the probe type—an echo probe or a path echo probe.
2. (Optional) Configure probe characteristics:
   - frequency
   - hops-of-statistics-kept (path echo)
   - max-response-failure (path echo)
   - operations-per-hop (path echo)
   - owner

- receive-interface
- request-data-size
- samples-of-history-kept
- tag
- timeout (echo)
- tos

NOTE: You cannot set any of these characteristics until you have set the probe type. The default values of these characteristics depend on the type of the entry.

3. (Optional) Set reaction conditions.
4. Schedule the probe.
5. (Optional) Capture statistics and collect error information.
6. (Optional) Collect history.

Configuring the Probe Type

To begin configuring RTR, enter RTR Configuration mode and configure the probe type—either an *echo* probe or a *path echo* probe.

rtr

- Use to configure an RTR probe and to enter RTR Configuration mode.
- Example

  host1(config)#**rtr 1**

- Use the **no** version to delete all configuration information for an RTR probe.

type

- Use to set an echo or path echo probe:
  - **echo**—Limited to end-to-end RTR operations; corresponds to SNMP ping
  - **pathEcho**—Finds a path to the destination and echoes each device in the path; corresponds to SNMP traceroute
- You must specify this value before any other.
- If you change the type for an existing RTR entry, all values are reset, including the administrative status. There is no default value.
- More than one RTR entry can become active, provided each entry's target address is unique.

- If you configure multiple RTR entries to use the same target address, you must issue the **receive-interface** command to specify the interface on which the RTR probe expects to receive responses. (For information, see *Setting the Receiving Interface* on page 68.)
- If you use a target address already configured for another RTR entry that is active, the test will not run if both entries are in the same virtual router. If they are in distinct virtual routers, however, there is no restriction.
- Example

  host1(config-rtr)#**type echo protocol ipIcmpEcho 10.10.0.9**

- Use the **no** version to remove the type configured for the probe.

Configuring Optional Characteristics

In addition to configuring the probe's type, you can configure the probe characteristics presented in Table 8.

Table 8: Probe Characteristics

Characteristic	Description
frequency	Time between tests (in seconds)
hops-of-statistics-kept	Hops per path for which statistics are gathered
max-response-failure	Maximum number of consecutive failures
operations-per-hop	Number of probes per hop
owner	Owner of the probe
receive-interface	Interface on which the probe expects to receive responses
request-data-size	Request's payload size
samples-of-history-kept	Maximum number of history samples
tag	User-defined tag
timeout	Probe timeout (in milliseconds)
tos	A value for the TOS byte

frequency

- Use to set the rate (in seconds) that the RTR probe uses to start a response time operation.
- Example

  host1(config-rtr)#**frequency 90**

- Use the **no** version to return to the default value, 60 seconds.

operations-per-hop

- Use to set the number of RTR probe operations sent to a given hop.
- You can apply this option only to a pathEcho type.
- Example

  host1(config-rtr)#**operations-per-hop 5**

- Use the **no** version to return to the default, 3.

owner

- Use to identify the owner of the probe.
- If the SNMP agent is the owner of the probe, the owner's name can begin with *agent*.
- Example

  host1(config-rtr)#**owner 192.10.27.6 rtc.boston.com 555.1212**

- Use the **no** version to return to the default, no owner.

request-data-size

- Use to set the protocol data size, in bytes, in the request packet.
- Example

  host1(config-rtr)#**request-data-size 20**

- Use the **no** version to return to the default value, 1 byte.

tag

- Use to set an identifier for the probe.
- Example

  host1(config-rtr)#**tag westford**

- Use the **no** version to return to the default, no tag.

timeout

- Use to set the time (in milliseconds) that the probe waits for a response.
- You can apply this option only to an echo type.
- Do not set the value for timeout to more than the value set for frequency. If you do, the timeout value is ignored.
- If you set the timeout to 0, no timeout is set.
- Example

  host1(config-rtr)#**timeout 3000**

- Use the **no** version to return to the default value, 5000 milliseconds.

tos

- Use to set the type of service (ToS) byte in the probe's IP header.
- Example

  host1(config-rtr)#**tos 16**

- Use the **no** version to return to the default value, 0. The default applies to both the echo and pathEcho types.

Capturing Statistics

The primary objective of RTR is to collect statistics and information about network performance. You can control the number and type of statistics collected.

hops-of-statistics-kept

- Use to set the number of hops per path for which statistics are collected.
- When the number of hops reaches the specified number (that is, *size*), no additional statistical information about the path is stored.
- This option applies only to pathEcho entries.
- To turn off this feature, set the value to 0.
- Example

  host1(config-rtr)#**hops-of-statistics-kept 5**

- Use the **no** version to set the default, 16 hops.

max-response-failure

- Use to set the maximum number of consecutive failures to respond to a probe's request.
- When the maximum number is reached, the test stops.
- This option applies only to pathEcho entries.
- To turn off this feature, set the value to 0.
- Example

  host1(config-rtr)#**max-response-failure 2**

- Use the **no** version to set the default, 5 consecutive failures.

Collecting History

RTR can collect data samples for a given probe. These samples are referred to as history data. When RTR collects history, it refers to tests. A test is the lifetime of a probe operation.

samples-of-history-kept

- Use to set the maximum number of entries in the history table for each RTR probe.
- This command enables you to control the number of samples saved in the history table.
- If you set the number of samples to 0, no samples are kept.
- Because collecting history increases memory usage, do so only when there is a problem in your network.
- Example

  host1(config-rtr)#**samples-of-history-kept 5**

- Use the **no** version to set the default, 16 hops for pathEcho type, 1 hop for echo type.

Setting the Receiving Interface

When you configure multiple RTR entries to use the same target address, you must issue the **receive-interface** command to set the interface on which the probe expects to receive responses. This action enables the router to map incoming responses to the proper RTR entry, even when multiple RTR entries have the same target address.

receive-interface

- Use to specify the interface on which the RTR probe expects to receive responses.
- You must set this attribute when multiple RTR entries are configured to use the same target address.
- Example

  host1(config-rtr)#**receive-interface fastEthernet 3/0**

- Use the **no** version to restore the default value, which is to receive a response on any interface.

Setting Reaction Conditions

You can set the RTR probe to react to events that take place and to send notifications about these events.

NOTE: The only **no** version for all the **rtr reaction-configuration** commands is **no rtr reaction-configuration** *rtrIndex*. Use the **no** version to clear all traps. This works for all the options.

rtr reaction-configuration action-type

- Use to specify the type of actions to occur depending on the events controlled by RTR.
- The default is to take the traps of enabled events.
- Example

  host1(config)#**rtr reaction-configuration 1 action-type trapOnly**

- There is no **no** version.

rtr reaction-configuration operation-failure

- Use to enable the operation-failure reaction.
- The operation-failure event is triggered when a number of consecutive probe operations are not received or when they are received after a timeout.
- Example

  host1(config)#**rtr reaction-configuration 1 operation-failure 3**

- There is no **no** version.

rtr reaction-configuration path-change

- Use to enable the path-change reaction.
- The path-change event is triggered when a change is detected in the hop table. At most, there can be one such event per test.
- Example

  host1(config)#**rtr reaction-configuration 1 path-change**

- There is no **no** version.

rtr reaction-configuration test-completion

- Use to enable test-completion reaction.
- The test-completion event is triggered when a test is completed successfully.
  - For echo, a successful test means that all probes were sent.
  - For pathEcho, a successful test means that the destination was reached at least once.
- At most, there can be one such event per test.
- Example

  host1(config)#**rtr reaction-configuration 1 test-completion**

- There is no **no** version.

rtr reaction-configuration test-failure

- Use to enable test-failure reaction.
- The test-failure event is triggered when a test fails. Failure is determined in the following ways:
  - If Echo, this event is triggered after testFailureValue probes are either not received or are received after a timeout.
  - If PathEcho, this event is triggered when the test ends and no responses are received from the destination.
- At most, there can be one such event per test.
- Example

  host1(config)#**rtr reaction-configuration 1 test-failure**

- There is no **no** version.

Scheduling the Probe

When you have configured the RTR probe, you must schedule the operation to begin collecting statistics and other information about problems that may arise.

rtr schedule

- Use to create an RTR schedule.
- Example

  host1(config)#**rtr schedule 5**

- Use the **no** version to stop the test. The **no** version stops the probe operation by putting it in the pending state. The **no** version also resets the restart-time attribute and the life attribute.

rtr schedule life

- Use to schedule the test's length.
- Life is a value that depends on the type of the RTR entry; it is not a length of time.
  - If the type is echo, life relates to the number of probes sent until a test finishes.
  - If the type is pathEcho, life relates to the maximum number of hops used by the traceRoute trap.
- Example

  host1(config)#**rtr schedule 5 life 1800**

- Use the **no** version to stop the test. The **no** version stops the probe operation by putting it in the pending state. The **no** version also resets the life attribute.

rtr schedule restart-time

- Use to specify a restart time, in seconds, after which a test is restarted.
- Example

  host1(config)#**rtr schedule 5 restart-time 15**

- Use the **no** version to stop the test. The **no** version stops the probe operation by putting it in the pending state. The **no** version also resets the restart-time attribute.

rtr schedule start-time

- Use to schedule a test's starting time (now or pending).
- Example

  host1(config)#**rtr schedule 5 start-time now**

- Use the **no** version to stop the test. The **no** version stops the probe operation by putting it in the default state, pending.

Shutting Down the Probe

You can shut down the RTR probe operation.

rtr reset

- Use to shut down the RTR, stop all probe operations, and clear the RTR configuration for the given virtual router.

NOTE: We recommend that you use this command only in extremely serious situations, such as problems with the configurations of a number of probe operations.

- Example

  host1(config)#**rtr reset**

- Use the **no** version to negate the reset operation.

Monitoring RTR

You can monitor RTR by displaying status and configuration information.

show rtr application

- Use to display global information about RTR.
- Field descriptions
  - numberOfEntries—Number of RTR entries according to type
  - entriesEnabled—RTR entries with administrative status enabled
  - entriesActive—RTR entries with operational status enabled
- Example

```
host1#show rtr application

          numberOfEntries      entriesEnabled     entriesActive
        ------------------   -----------------  ---------------
echo                    1                   1               1
pathEcho                1                   1               1
total                   2                   2               2
```

show rtr collection-statistics

- Use to display statistical information for a particular probe operation or for all operations.
- Field descriptions
  - rtrIndex—Index number of the RTR probe
  - operationsSent—Number of probe operations sent
  - operationsRecvd—Number of probe operations received
  - lastGoodResponse—Time when last valid probe operation was received
  - operStatus—Operational status of the probe: enabled, disabled
  - minRtt—Minimum round-trip time in milliseconds
  - maxRtt—Maximum round-trip time in milliseconds
  - avgRtt—Average round-trip time in milliseconds
  - rttSumSqr—Sum of the square of all round-trip times in milliseconds
  - testAttempts—Number of times the test ran
  - testSuccesses—Number of times the test ran successfully
  - currentHop—Current hop (TTL) used in the test
  - currentOperation—Current probe operation index sent to the hop
- Example

```
host1#show rtr collection-statistics

  Echo Entries:

  rtrIndex  operationsSent  operationsRcvd  lastGoodResponse
----------  --------------  --------------  ----------------
     1           5208            5187       08/30/2000 05:09
```

```
  rtrIndex  operStatus  minRtt  maxRtt  avgRtt  rttSumSqr
----------  ----------  ------  ------  ------  ---------
     1          enabled   0      1785     3     7109208

  PathEcho Entries:

  rtrIndex  testAttempts  testSuccesses  lastGoodResponse
----------  ------------  -------------  ----------------
     2           156           156       08/30/2000 05:09

  rtrIndex  operStatus  currentHop  currentOperation
----------  ----------  ----------  -----------------
     2       enabled        2             4
```

show rtr configuration

- Use to display the configuration for a particular probe or for all probes.
- Field descriptions
  - rtrIndex—Index number of the RTR probe
  - type—Probe type: echo, pathEcho
  - targetAddress—Address of the probe's target
  - reqSize—Protocol data size in the request packet
  - freq—Rate in seconds that the RTR probe uses to start a response time operation
  - life—Length of the test
  - source—Interface from which the probe is sent
  - restartTime—Restart time of the test in seconds
  - owner—Owner of the probe
  - samples—Maximum number of entries saved in the history table for this RTR probe
  - admin—Administrative status of the probe: enabled, disabled
  - tos—Setting of the type of service (ToS) byte in the probe's IP header
  - reactionConfiguration—RTR reactions that are configured for the probe
  - receiveInterface—Type and specifier of the interface on which the probe expects to receive responses; this field is blank if the optional **receive-interface** characteristic is not configured
  - operFail—Operation failure event is triggered when this number of consecutive probe operations is not received or when the operations are received after a timeout
  - testFail—Test failure event is triggered when this number of probe operations is not received or when the operations are received after a timeout
  - timeout—Time in milliseconds that the probe waits for a response
  - tag—Identifier configured for the probe

- operPerHop—Number of RTR probe operations sent to a given hop
- maxFail—Maximum number of consecutive failures to respond to a probe's request. When the maximum number is reached, the test stops. Applies only to pathEcho entries.
- hopKpt—Number of hops per path for which statistics are collected. When this number is reached, no additional statistical information about the path is stored. Applies only to pathEcho entries.

- Example

```
host1#show rtr configuration
  rtrIndex     type   targetAddress reqSize freq   life
  -------  --------   ------------  ------  ------ -----
      1       echo    10.5.0.200       1     1       20
      2   pathEcho    10.5.0.11        1     1       30

  rtrIndex     source             restartTime  owner
  ----------  ------------------  -----------  ----------
      1       fastEthernet0/0         10
      2                               60

  rtrIndex    samples  admin     tos  reactionConfiguration
  ----------  -------  --------  ---  -------------------------
      1          5     enabled    0
      2          5     enabled    0

  rtrIndex    receiveInterface
  ----------  ----------------
      1       fastEthernet0/0

  rtrIndex    operFail  testFail  timeout  tag
  ----------  --------  --------  -------  ------
      1          1         1      10000

  rtrIndex  operPerHop  maxFail  hopKpt  tag
----------  ----------  -------  ------  ------
      2          5         3      16
```

show rtr history

- Use to display history (data samples) for a particular probe or for all probes.
- Field descriptions
  - rtrIndex—Index number of the RTR probe
  - operation—Index number of the probe operation
  - rtt—Round-trip time in milliseconds
  - statusDescription
    - concurrentLimitFail—Target already being used by another rtrIndex
    - ifInactiveToTarget—Interface used to reach target is not operational
    - invalidHostAddress—Target address is not supported
    - noRouteToTarget—Target address is not reachable
    - responseReceived—Probe operation replied by target

  - requestTimedOut—Probe operation not replied to by target or reply received after timeout
  - unknownDestAddress—Target address is invalid
  - unableToResolveName—Target address could not be looked up
- timeStamp—Date and time when the RTR entry was created
- test—Index number of the pathEcho test
- hop—Index number of the hop count
- operation—Index number of the probe operation
- address—Address of router at the hop

- Example

```
host1#show rtr history
 Echo Entries:

  rtrIndex   operation  rtt   statusDescription timeStamp
  --------   ---------  ---   ----------------- ---------
         1        5476  0     responseReceived  08/30/2000 05:17
         1        5477  0     responseReceived  08/30/2000 05:17
         1        5478  0     responseReceived  08/30/2000 05:17
         1        5479  0     responseReceived  08/30/2000 05:17
         1        5480  0     responseReceived  08/30/2000 05:17

  PathEcho Entries:

  rtrIndex    test    hop   operation  rtt     statusDescription
  ----------  ------  ----  ---------  ------  -----------------
           2     165     3          5       0  responseReceived
           2     165     3          1       0  responseReceived
           2     165     3          2       0  responseReceived
           2     165     3          3       0  responseReceived
           2     165     3          4       0  responseReceived

 rtrIndex    test    hop   operation  timeStamp         address
 ----------  ------  ----  ---------  ----------------  --------
          2     165     3          5  08/30/2000 20:39  10.5.0.11
          2     165     3          1  08/30/2000 20:40  10.5.0.11
          2     165     3          2  08/30/2000 20:40  10.5.0.11
          2     165     3          3  08/30/2000 20:40  10.5.0.11
          2     165     3          4  08/30/2000 20:40  10.5.0.11
```

show rtr hops

- Use to display RTR hops information.
- Field descriptions
  - rtrIndex—Index number of the RTR probe
  - hop—Index number of the hop count
  - address—Address of the router at the hop
  - minRtt—Minimum round-trip time in milliseconds
  - maxRtt—Maximum round-trip time in milliseconds
  - avgRtt—Average round-trip time in milliseconds
  - rttSumSqr—Sum of the square of all round-trip times in milliseconds

- operationsSent—Number of probe operations sent
- operationsRcvd—Number of probe operations received
- lastGoodResponse—Time when last valid probe operation was received

- Example

```
host1#show rtr hops

  rtrIndex    hop   address     minRtt  maxRtt  avgRtt  rttSumSqr
  ----------  ----  ----------- ------  ------  ------  --------
           2     1  192.168.1.1      1     276       1     955363
           2     2  10.2.0.3         0    1109       2   10094451

  rtrIndex hop operationsSent operationsRcvd  lastGoodResponse
  -------- --- -------------- --------------  ------------
         2   1          36985          36838  09/18/2000 20:20
         2   2          30717          21494  09/18/2000 20:20
```

show rtr operational-state

- Use to display RTR operational information.
- Field descriptions
  - rtrIndex—Index number of the RTR probe
  - type—Type of RTR probe: echo, pathEcho
  - entryStatus—If the entry was created via the SNMP DISMAN MIB, the row may be partially constructed; if that is the case, the CLI displays notReady as the entry's status
  - adminStatus—Derived from the **rtr schedule start-time** command; if the option is **now**, the status is enabled; if the option is **pending**, the status is disabled
  - operStatus—Enabled only if entryStatus and adminStatus are enabled and the test is running; operStatus remains enabled if the test finishes and restart time is not 0
- Example

```
host1#show rtr operational-state
 rtrIndex    type      entryStatus  adminStatus  operStatus
 ----------  --------  -----------  -----------  ----------
          1  echo      active       enabled      enabled
          2  pathEcho  active       enabled      enabled
```

Monitoring IP

This section explains how to set a statistics baseline and use the **show** commands to view your IP configuration and monitor IP interfaces and statistics.

System Event Logs

To troubleshoot and monitor IP, use the following system event logs:

- ipAccessList—IP access list matching
- ipEngine—IP chassis manager
- ipGeneral— IP general information
- ipIfCreator—IP interface creator events
- ipInterface—IP interface events
- ipNhopTrackerGeneral—Next-hop tracker for IP shared interfaces
- ipProfileMgr—IP profile manager events
- ipRouteTable—IP routing table events
- ipTraffic—IP frame transmit and receive events
- ipTunnel—IP tunnel events

For more information about using event logs, see the *JUNOSe System Event Logging Reference Guide, Chapter 1,* .

Establishing a Baseline

IP statistics are stored in system counters. The only way to reset the system counters is to reboot the router. You can, however, establish a baseline for IP statistics by setting a group of reference counters to zero.

baseline ip

- Use to set a statistics baseline for IP statistics. Baselining is not supported for IP socket statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ip**

- There is no **no** version.

baseline ip udp

- Use to set a statistics baseline for UDP statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ip udp**

- There is no **no** version.

baseline tcp

- Use to set a statistics baseline for all (both IPv4 and IPv6) TCP statistics or for only IPv4 or IPv6 statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **ip** keyword to implement a baseline for only IPv4 statistics.
- Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.
- Example 1

  host1#**baseline tcp**

- Example 2

  host1#**baseline ip tcp**

- There is no **no** version.

IP show Commands

You can monitor the following aspects of IP using **show ip** commands:

To Display	Command
Access lists	show access-list show ip as-path-access-list
ARP	show arp
General IP information	show ip
IP addresses	show ip address
Community lists	show ip community-list
Routing table	show ip forwarding-table slot show forwarding-table route-holddown
Interfaces	show ip interface
Shared IP interfaces	show ip interface shares

To Display	Command
Protocols	**show ip protocols**
Redistribution policies	**show ip redistribute**
Routes	**show ip route**
Interfaces and next hops	**show ip route slot**
Socket statistics	**show ip socket statistics**
Static routes	**show ip static**
TCP ACK, RST, and SYN protection status	**show tcp ack-rst-and-syn**
Black hole threshold information	**show tcp path-mtu-discovery**
TCP statistics	**show tcp statistics**
Traffic	**show ip traffic**
UDP statistics	**show ip udp statistics**
Profiles	**show ip profile**
Route maps	**show route-map**

To set a statistics baseline for IP interfaces, use the **baseline tcp** and **baseline ip udp** commands. Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string that you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

show access-list

- Use to display information about access lists, including the instances of each access list.
- Example

```
host1#show access-list
IP Access List 1:
    permit ip 172.31.192.217 0.0.0.0 0.0.0.0 255.255.255.255
    permit ip 12.40.0.0 0.0.0.3 0.0.0.0 255.255.255.255
    deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
IP Access List 2:
    permit ip 172.19.0.0 0.0.255.255 0.0.0.0 255.255.255.255
    deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
IP Access List 10:
    permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
IP Access List 11:
    deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

show arp

- Use to display information about ARP.
- Field descriptions
  - Address—IP address of the entry
  - Age—Time to live for this entry in seconds
  - Hardware Addr—Physical (MAC) address of the entry
  - Interface—Interface-specifier of the entry (for example, fastEthernet6/0 is an Ethernet interface on slot 6, port 0)
  - *—Indicates that an ARP entry was added because of an **arp validate** command, rather than just an **arp** command.
- Example

```
host1#show arp
        Address     Age   Hardware Addr  Interface
 172.31.192.217   21340  00d0.58f2.67e0 loopback1
    192.168.1.0   20730  00e0.09ed.5312 fastEthernet6/0 *
    192.168.1.1   12550  00e0.b06a.4c75 fastEthernet6/0 *
  192.168.1.217   21600  0090.1a00.0230 fastEthernet6/0 *
  192.168.1.255   21600  00f0.c2d1.1200 fastEthernet6/0 *
      12.40.0.2   24320  0020.6393.4233 atm5/0.1
     172.18.2.1   21600  0020.bed2.8738 atm5/1.1
     172.18.2.2   21600  0020.5b91.60f2 atm5/1.1
 172.31.192.206   21600  00d0.43b5.1032 atm5/1.1
```

show forwarding-table route-holddown

- Use to display the configured hold-down time allotted after an initial routing table change for the accumulation and subsequent distribution of a set of routing table updates to the line modules. The default value is 3 seconds; the range of values is 0–30 seconds.
- A higher hold-down setting can enhance SRP performance; however, a higher setting can also delay the implementation of routing table changes on the line modules.
- A hold-down timer value of zero (0) distributes an update after each change to the routing table.
- Example

```
host1#show forwarding-table route-holddown
  Hold-down timer value is 3 seconds.
```

show ip

- Use to display general information about IP.
- Field descriptions
  - IP Router Id—Router ID number
  - Router Name—Router name
  - Default TTL—Default IP TTL (time-to-live) value
  - Reassemble Timeout—Amount of time (in minutes) IP waits for missing packet fragments before it drops the fragments it is holding
  - SA Validate Trap—Whether the source address validation trap is enabled
- Example

```
host1#show ip
  IP Router Id: 192.168.1.155
  Router Name: default
  Default TTL: 60
  Reassemble Timeout: 30
  SA Validate Trap: false
```

show ip address

- Use to display detailed or summary information about a particular IP interface.
- Specify a VRF name to view information for only that VRF.
- Use the **brief** keyword to display summary information about the interface.
- Use the **detail** keyword to display detailed information about the interface.
- Field descriptions
  - Network Protocols—Network protocols configured on this interface
  - Internet address—IP address and subnet mask of this interface
  - Broadcast address—Broadcast address of this interface
  - Operational MTU—MTU of this interface
  - Administrative MTU—Value of the MTU if it has been administratively overridden using the configuration
  - Operational speed—Speed of the interface
  - Administrative speed—Value of the speed if it has been administratively overridden using the configuration
  - Discontinuity Time—Value of the SysUpTime when the interface statistics last started being valid
  - Router advertisement—Status of router discovery advertisement: enabled, disabled
  - Proxy Arp—Status of the feature: enabled, disabled
  - Administrative debounce-time—Configured debounce behavior, enabled or disabled. If enabled, indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. If the state changes back before the debounce timer expires, no state change is reported.

- Operational debounce-time—Current debounce behavior, enabled or disabled. If enabled, indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. If the state changes back before the debounce timer expires, no state change is reported.
- Access routing—Access route addition: enabled, disabled
- Multipath mode—Equal cost multipath mode method: hashed, round-robin
- In Received Packets, Bytes—Total number of packets and bytes received on this interface
  - Unicast Packets, Bytes—Unicast packets and bytes received on the IP interface; link-local received multicast packets (non-multicast-routed frames) are counted as unicast packets
  - Multicast Packets, Bytes—Multicast packets and bytes received on the IP interface which are then multicast-routed are counted as multicast packets
- In Policed Packets, Bytes—Packets and bytes that were received and dropped because of rate limits
- In Error Packets—Number of packets received with errors
- In Invalid Source Address Packets—Packets received with invalid source address (for example, spoofed packets)
- In Discarded Packets—Packets received that were discarded for reasons other than rate limits, errors, and invalid source address
- Out Forwarded Packets, Bytes—Total number of packets and bytes that were sent from this interface
  - Unicast Packets, Bytes—Unicast packets and bytes that were sent from this interface
  - Multicast Routed Packets, Bytes—Multicast packets and bytes that were sent from this interface
- Out Scheduler Drops Committed Packets, Bytes—Outgoing packets and bytes dropped by the scheduler even though they had a committed traffic contract
- Out Scheduler Drops Conformed Packets, Bytes—Outgoing packets and bytes dropped by the scheduler even though they conformed to the traffic contract
- Out Scheduler Drops Exceeded Packets, Bytes—Outgoing packets and bytes that were dropped by the scheduler because they exceeded the contract
- Out Policed Packets, Bytes—Outgoing packets and bytes dropped because of rate limiters
- Out Discarded Packets—Outgoing packets that were discarded for reasons other than those dropped by the scheduler and those dropped because of rate limits

- Example

```
host1#show ip address 10.6.136.73
fastEthernet0/0 is up, line protocol is up
  Network Protocols: IP
  Internet address is 10.6.136.73/255.255.128.0
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 1  Administrative speed = 0
  Discontinuity Time = 5766
  Router advertisement = disabled
  Proxy Arp = disabled
  Administrative debounce-time = 10 mSecs
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 2849, Bytes 759428
    Unicast Packets 2849, Bytes 759428
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 1866, Bytes 84650
    Unicast Packets 1866, Bytes 84650
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0
```

show ip as-path-access-list

- Use to display information about AS-path access lists.
- Example

```
host1#show ip as-path-access-list
AS Path Access List 1:
    permit  .*
AS Path Access List 2:
    deny  .*
AS Path Access List 3:
    permit  _109_
    deny  .*
AS Path Access List 4:
    permit  _109$
    deny  .*
AS Path Access List 10:
    deny  _109$
    permit  ^108_
    deny  .*
```

show ip community-list

- Use to display routes that are permitted by a BGP community list.
- Example

```
host1#show ip community-list
Community List 1:
  permit 752877569 (11488:1)
  permit 752877570 (11488:2)
  permit 752877571 (11488:3)
  permit 752877572 (11488:4)
Community List 2:
  permit 4294967043 (local-as)
```

show ip forwarding-table slot

- Use to display details on the forwarding table for a specific line module, including the memory used by each virtual router configured on the line module and free memory available on the module.
- The Load Errors field records any failed routing table distribution attempt as an error. Attempts can fail for many reasons during normal operation; a failed attempt does not necessarily indicate a problem. It is normal to see many Load Errors per day.
- If the Status field does not indicate Valid, then the routing table distribution has failed constantly for that VR. It is normal and appropriate behavior for the Status field to indicate Valid while the Load Error field increases daily.
- Field descriptions
  - Free Memory—Amount of routing table memory free on the line module, in kilobytes
  - Virtual Router—Name of the virtual routers configured on the line module
  - Memory (KB)—Amount of routing table memory consumed by the virtual router, in kilobytes
  - Load Errors—Count of errors made while loading the routing table on the line module
  - Status—Whether the routing table for the virtual router is valid
- Example

```
host1#show ip forwarding-table slot 9
Free Memory = 3,166KB
  Virtual Router      Memory       Load Errors     Status
                      (KB)
----------------     ---------    -------------    --------
   vr1                 4128             0          Valid
   vr2                 3136             0          Valid
   vr3                 2256             0          Valid
default               1024              0         Valid
------------------------------------------------------------
```

show ip interface

- Use to display the current state of all IP interfaces or the IP interfaces you specify.
- The default is all interface types and all interfaces.
- The **show-virtual-router-keyword** displays virtual router information.
- Field descriptions
  - interface—Interface type and interface specifier
  - interface status—Status of the interface
  - line protocol—Status of the line protocol
  - Description—Text description or alias if configured for the interface
  - Link up/down trap—Status of SNMP link up/down traps on the interface
  - Internet address—IP address of the interface
  - IP Statistics Rcvd:
    - local destination—Frames with this router as their destinations
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IP Statistics Frags:
    - reasm ok—Number of reassembled packets
    - reasm req—Number of requests for reassembly
    - reasm fails—Number of reassembly failures
    - frag ok—Number of packets fragmented successfully
    - frag req—Number of frames requiring fragmentation
    - frag fails—Number of packets unsuccessfully fragmented
  - IP Statistics Sent:
    - generated—Number of packets generated
    - no routes—Number of packets that could not be routed
    - discards—Number of packets that could not be routed that were discarded
  - ICMP Statistics Rcvd:
    - errors—Error packets received
    - dst unreach—Packets received with destination unreachable
    - time exceed—Packets received with time-to-live exceeded
    - param probs—Packets received with parameter errors
    - src quench—Source quench packets received
    - redirect—Receive packet redirects

- echo req—Echo request (ping) packets
- echo rpy—Echo replies received
- timestamp req—Requests for a timestamp
- timestamp rpy—Replies of timestamp requests
- addr mask req—Mask requests sent
- addr mask rpy—Mask replies sent

- ICMP Statistics Sent:
  - errors—Error packets sent
  - dst unreach—Packets sent with destination unreachable
  - time excd—Packets sent with time-to-live exceeded
  - param probs—Packets sent with parameter errors
  - src quench—Source quench packets sent
  - redirect—Send packet redirects
  - timestamp req—Requests for a timestamp
  - timestamp rpy—Replies to timestamp requests
  - addr mask req—Address mask requests
  - addr mask rpy—Address mask replies
- In Received Packets, Bytes—Total number of packets and bytes received on the IP interface
  - Unicast Packets, Bytes—Unicast packets and bytes received on the IP interface; link-local received multicast packets (non-multicast-routed frames) are counted as unicast packets
  - Multicast Packets, Bytes—Multicast packets and bytes received on the IP interface which are then multicast-routed are counted as multicast packets
- In Forwarded Packets, Bytes—Packets and bytes forwarded into an output IP interface
- In Total Dropped Packets, Bytes—Total number of packets and bytes that were dropped on the interface; sum of all the drop reasons indented below this field
  - In Policed Packets—Packets discarded on a receive IP interface because of token bucket limiting, a drop action in a policy, or discarded MAC validation packets
  - In Invalid Source Address Packets—Packets discarded on a receive IP interface due to invalid IP source address (sa-validate enabled)
  - In Error Packets—Packets discarded on a receive IP interface due to IP header errors
  - In Discarded Packets—Packets discarded on the ingress interface due to a configuration problem rather than a problem with the packet itself
  - In Fabric Dropped Packets—Packets discarded on a receive IP interface due to internal fabric congestion

- Out Forwarded Packets, Bytes—Total number of packets and bytes forwarded out the IP interface
  - Unicast Packets, Bytes—Unicast packets and bytes forwarded out the IP interface
  - Multicast Routed Packets, Bytes—Multicast packets and bytes forwarded out the IP interface
- Out Requested Packets, Bytes—Packets and bytes requested to be forwarded out an IP interface
- Out Total Dropped Packets, Bytes—Total number of packets and bytes that were discarded on the egress interface; sum of all the drop reasons indented below this field
  - Out Scheduler Drops Committed Packets, Bytes—Packets and bytes dropped by the scheduler even though they had a committed traffic contract
  - Out Scheduler Drops Conformed Packets, Bytes—Packets and bytes dropped by the scheduler even though they conformed to the traffic contract
  - Out Scheduler Drops Exceeded Packets, Bytes—Packets and bytes dropped by the scheduler because they exceeded the contract
  - Out Policed Packets—Packets discarded on the egress interface due to rate limiting
  - Out Discarded Packets—Packets discarded on the egress interface due to a configuration problem rather than a problem with the packet itself
  - Out Fabric Dropped Packets—Packets dropped due to internal fabric congestion

- Example

```
host1#show ip interface detail fastEthernet 0/0
fastEthernet0/0 is up, line protocol is up
  Description: boston00 fast ethernet interface
  Link up/down trap is disabled

  Internet address is 1.1.1.2/255.255.255.0
IP statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Frags: 0 reasm ok, 0 reasm req, 0 reasm fails
         0 frag ok, 0 frag creates, 0 frag fails
  Sent:  31656835 generated, 0 no routes, 0 discards
ICMP statistics:
  Rcvd:  0 errors, 0 dst unreach, 0 time exceed
         0 param probs, 0 src quench, 0 redirect,
         0 echo req, 31656816 echo rpy
         0 timestmp req, 0 timestmp rpy
         0 addr mask req, 0 addr mask rpy
  Sent:  0 errors, 0 dst unreach, 0 time excd
         0 param probs, 0 src qnch, 0 redirect
         0 timestamp req, 0 timestamp rpy
         0 addr mask req, 0 addr mask rpy
In Received Packets 246220, Bytes 344624800
    Unicast Packets 246162, Bytes 344621410
    Multicast Packets 58, Bytes 3390
```

```
In Forwarded Packets 245464, Bytes 343566400
In Total Dropped Packets 756, Bytes 1058400
    In Policed Packets 756
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0
    In Fabric Dropped Packets 0

  Out Forwarded Packets 117, Bytes 87297
    Unicast Packets 117, Bytes 87297
    Multicast Routed Packets 0, Bytes 0
  Out Requested Packets 117, Bytes 87297
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Drops Committed Packets 0, Bytes 0
    Out Scheduler Drops Conformed Packets 0, Bytes 0
    Out Scheduler Drops Exceeded  Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0
    Out Fabric Dropped Packets 0
```

If you are losing packets because of fabric congestion, you can use the In Fabric Dropped Packets and Out Fabric Dropped Packets statistics to help determine the location of the bottleneck. Both statistics count the same thing—the same packets dropped because of fabric congestion—but in different directions.

At any given time, the total number of packets dropped in the fabric for all interfaces in the chassis is equal to the sum of all In Fabric Dropped Packets for all interfaces in the chassis, which equals the sum of all Out Fabric Dropped Packets for all interfaces in the chassis.

Packets not dropped for another listed reason are considered to have been dropped in the fabric. The router calculates In Fabric Dropped Packets by subtracting the total number of inbound packets dropped for all other reasons from the In Total Dropped Packets number. The router calculates Out Fabric Dropped Packets by subtracting the total number of outbound packets dropped for all other reasons from the Out Total Dropped Packets number.

The router calculates In Total Dropped Packets by subtracting In Forwarded Packets from In Received Packets. The router calculates Out Total Dropped Packets by subtracting Out Forwarded Packets from Out Received Packets. These statistics are reported while traffic is moving through the router. The router can get false statistics based on packets being forwarded or received after polling and based on which of the statistics is reported first. For example, In Forwarded Packets can be reported as greater than In Received Packets. Rather than displaying In Total Dropped Packets as a negative value, the command displays it as the sum of all drop reasons other than fabric drops; fabric drops are reported as 0, but might actually be nonzero. If you halt traffic, the In Total Dropped Packets and Out Total Dropped Packets values are always correct.

show ip interface shares

- Use to display information about shared IP interfaces.
- If you specify an IP interface specifier, the command displays information only for that interface and any shared IP interfaces associated with it.
- Field descriptions
  - Interface—Interface specifier or name of the interface
  - IP-Address—IP address associated with the interface
  - Status—Operational state of the interface
  - Protocol—State of the protocol running on the interface
  - Virtual Router—Virtual router in which the interface is configured
- Example 1

```
host1#show ip interface shares brief

Interface        IP-Address         Status    Protocol   Virtual Router
null0            255.255.255.255/32  up         up
fastEthernet0/0  10.13.5.17/24       up         up
loopback100      202.1.1.1/24        up         up
atm4/0.1         10.1.1.1/24         up         up
ip si0           Unnumbered          up         up        vr-a
ip si1           Unnumbered          up         up        vr-b:vrf-1
```

- Example 2

```
host1#show ip interface shares brief atm 4/0.1

Interface        IP-Address         Status    Protocol   Virtual Router
atm4/0.1         10.1.1.1/24         up         up
ip si0           Unnumbered          up         up        vr-a
ip si1           Unnumbered          up         up        vr-b:vrf-1
```

- Example 3—For a description of the following fields, see the **show ip address** command

```
host1#show ip interface shares atm 4/0.1
atm4/0.1 is up, line protocol is up
  Network Protocols: IP
  Unnumbered Interface on loopback100
  ( IP address  202.1.1.1 )
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 120, Bytes 12000
    Unicast Packets 60, Bytes 6000
    Multicast Packets 60, Bytes 6000
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  Out Forwarded Packets 101, Bytes 5252
    Unicast Packets 101, Bytes 5252
    Multicast Routed Packets 0, Bytes 0
```

```
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0

ip si0 is up, line protocol is up
  Network Protocols: IP
  Virtual Router vr-a
  Layer 2 interface atm4/0.1
  Unnumbered Interface on loopback100
  ( IP address  202.1.1.1 )
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  Out Forwarded Packets 101, Bytes 5252
    Unicast Packets 101, Bytes 5252
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0

ip si1 is up, line protocol is up
  Network Protocols: IP
  Virtual Router vr-b:vrf-1
  Layer 2 interface atm4/0.1
.
.
.
  Out Policed Packets 0, Bytes 0
```

- Example 4

```
host1#show ip interface shares ip si0
ip0 is up, line protocol is up
  Network Protocols: IP
  Layer 2 interface atm4/0.1
  Unnumbered Interface on loopback100
  ( IP address  202.1.1.1 )
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  Out Forwarded Packets 101, Bytes 5252
    Unicast Packets 101, Bytes 5252
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
```

show ip profile

- Use to display information about a specific IP profile.
- Field descriptions
  - IP profile—Profile name
  - IP address—IP address and subnet mask of the interface or none if the interface is unnumbered
  - Unnumbered interface—Specifier for the unnumbered interface or none if the interface is numbered
  - Router—Router name
  - Directed Broadcast—Enabled or disabled
  - ICMP Redirects—Enabled or disabled
  - Access Route Addition—Enabled or disabled
  - Network Address Translation—Enable or disable; domain location (inside or outside)
  - Source-Address Validation—Enabled or disabled
  - Ignore DF Bit—Enabled or disabled
  - Administrative MTU—MTU size

- Auto Detect—Router automatically detects packets that do not match any entries in the demultiplexer table; enabled or disabled
- Auto Configure—Dynamic creation of subscriber interfaces on a primary IP interface; enabled or disabled
- IP FlowStats—Enabled or disabled

- Example

```
host1#show ip profile foo
IP profile : foo
IP address                  : none
Unnumbered interface        : none
Router                      :
Directed Broadcast          : Enabled
ICMP Redirects              : Disabled
Access Route Addition       : Enabled
Network Address Translation: Enabled, domain inside
Source-Address Validation   : Enabled
Ignore DF Bit               : Disabled
Administrative MTU          : 0
Auto Detect                 : Disabled
Auto Configure              : Disabled
Auto Detect                 : Disabled
IP FlowStats                : Enabled
```

show ip protocols

- Use to display configured protocols.
- Field descriptions
  - For BGP:
    - Redistributing—Protocol to which BGP is redistributing routes
    - Default local preference—Local preference value
    - IGP synchronization—Status of IGP synchronization: enabled, disabled
    - Always compare MED—Status of multiexit discrimination: enabled, disabled
    - Router flap damping—Status of route dampening: enabled, disabled
    - Administrative Distance—External, internal, and local administrative distances
    - Neighbor Address—IP address of the BGP neighbor
    - Neighbor Incoming/Outgoing update distribute list—Number of the access list for outgoing routes
    - Neighbor Incoming/Outgoing update prefix list—Number of the prefix list for incoming or outgoing routes
    - Neighbor Incoming/Outgoing update prefix tree—Number of the prefix tree for incoming or outgoing routes
    - Neighbor Incoming/Outgoing update filter list—Number of filter list for incoming routes
    - Routing for Networks—Network for which BGP is currently injecting routes

- For IS-IS:
  - System Id—6-byte value of the system
  - IS-Type—Routing type of the router: Level 1, Level 2
  - Distance—Administrative distance for IS-IS learned routes
  - Address Summarization—Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned from other routing protocols
  - Routing for Networks—Network for which IS-IS is currently injecting routes
- For OSPF:
  - Router ID—OSPF process ID for the router
  - Distance—Administrative distance for OSPF learned routes
  - Redistributing—Protocol to which OSPF is redistributing routes
  - Address Summarization—Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned from other routing protocols
  - Routing for Networks—Network for which OSPF is currently injecting routes
- For RIP:
  - Router Administrative State—RIP protocol state. Enable means that the interface is allowed to send and receive updates. Disable means that the interface, if it is configured, is not enabled to run yet.
  - System version—RIP versions allowed for sending and receiving RIP updates. The router version is currently set to RIP1, which sends RIP version 1 but will receive version 1 or 2. If the version is set to RIP2, the router will send and receive version 2 only. The default is configured for RIP1.
  - Update interval—Current setting of the update timer (in seconds)
  - Invalid after—Current setting of the invalid timer (in seconds)
  - hold down time—Current setting of the hold down timer (in seconds)
  - flushed interval—Current setting of the flush timer (in seconds)
  - Filter applied to outgoing route update—Access list applied to outgoing RIP route updates
  - Filter applied to incoming route update—Access list applied to incoming RIP route updates
  - Global route map—Route map that specifies all RIP interfaces on the router
  - Distance—Value added to RIP routes added to the IP routing table; the default is 120.

- Interface—Interface type on which RIP protocol is running
- Redistributing—Protocol to which RIP is redistributing routes
- Routing for Networks—Network for which RIP is currently injecting routes

- Example

```
host1#show ip protocols
Routing Protocol is “bgp 100”
  Redistributing: ospf
  Default local preference is 100
  IGP synchronization is enabled
  Always compare MED is disabled
  Router flap damping is disabled
  Administrative Distance: external 20 internal 200 local 200
  Neighbor(s):
    Address 1.1.1.1
    Outgoing update distribute list is 2
    Outgoing update prefix list is efg
    Incoming update prefix tree is abc
    Incoming update filter list is 1
  Routing for Networks:
    192.168.1.0/24

Routing Protocol is “isis isisOne”
  System Id: 0000.0000.0011.00  IS-Type: level-1-2
  Distance: 115
  Address Summarization:
    None
  Routing for Networks:
    fastEthernet0/0

Routing Protocol is “ospf 1” with Router ID 192.168.1.151
  Distance is 110
  Redistributing: isis
  Address Summarization:
    None
  Routing for Networks:
    192.168.1.0/255.255.255.0 area 0.0.0.0

Routing Protocol is “rip”
  Router Administrative State: enable
  System version RIP1: send = 1, receive = 1 or 2
  Update interval: 30 seconds
  Invalid after: 180  seconds
  hold down time: 120 seconds
  flushed interval: 300 seconds
  Filter applied to outgoing route update is not set
  Filter applied to incoming route update is not set
  No global route map
  Distance is 120
  Interface         Tx   Rx   Auth
  fastEthernet0/0   1    1,2  none
  Redistributing: ospf
  Routing for Networks:
    192.168.1.0/255.255.255.0
```

show ip redistribute

- Use to display configured route redistribution policy.
- Field descriptions
  - To—Protocol that routes are distributed into
  - From—Protocol that routes are distributed from
  - status—Redistribution status
  - route map number—Number of the route map
- Example

```
host1#show ip redistribute
To ospf, From static is enabled with route map 4
To ospf, From connected is enabled with route map 3
```

show ip route

- Use to display the current state of the routing table, including routes not used for forwarding.
- You can display all routes, a specific route, best route to a resolved domain name, all routes beginning with a specified address, routes for a particular protocol (BGP, IS-IS, OSPF, or RIP), locally connected routes, internal control routes, static routes, or summary counters for the routing table.
- Field descriptions
  - Protocol/Route type codes—Protocol and route type codes for the table that follows
  - Prefix—IP address prefix of network destination
  - Length—Network mask length for prefix
  - Next Hop—IP address of the next hop to the route, whether it is a local interface or another router
  - Dist—Administrative distance for the route; see Table 6
  - Met—Number of hops
  - Intf—Interface type and interface specifier
- Example 1

```
host1#show ip route
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VR/VRF, *- indirect next-hop

  Prefix/Length    Type    Next Hop       Dist/Met   Intf
  -------------    ----    --------       --------   ------
  172.16.2.0/24     Bgp     192.168.1.102   20/1       fastEthernet0/0
  10.10.0.112/32    Static  192.168.1.1     1/1        fastEthernet0/0
  10.1.1.0/24       Connect 10.1.1.1        0/1        atm3/0.100
```

- Example 2

```
host1#show ip route static
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VR/VRF, *- indirect next-hop

 Prefix/Length  Type   Next Hop   Dist/Met  Intf
 -------------  ----   --------   --------  --------------
10.10.0.112/32  Static 192.168.1.1  1/1     fastEthernet0/0
```

- Example 3

```
host1#show ip route summary
5 total routes, 720 bytes in route entries
0 isis routes
0 rip routes
1 static routes
1 connected routes
0 bgp routes
0 ospf routes
3 other internal routes
0 access routes
0 internally created access host routes

Last route added/deleted: 0.0.0.0/0 by StaticLow
At THU MAR 09 2000 05:22:49 UTC
```

- Example 4

```
host1#show ip route all
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VR/VRF, *- indirect next-hop

  Prefix/Length      Type     Next Hop          Dist/Met    Intf
  -------------      ----     --------          --------    ------
  0.0.0.0/0        Static  192.168.1.1      1/1        fastEthernet0/0
1.1.1.1/32       I2-E-i  192.168.1.105    115/10     fastEthernet0/0
6.6.6.0/24       Static  192.168.1.1      1/1        fastEthernet0/0
6.33.5.0/24      Static  0.0.0.0          1/1        loopback2
8.8.8.0/24       I2-E-i  192.168.1.105    115/10     fastEthernet0/0
9.9.9.9/32       I2-E-i  192.168.1.105    115/10     fastEthernet0/0
10.0.0.0/8       I2-E-i  192.168.1.105    115/10     fastEthernet0/0
10.10.0.156/32   Static  192.168.1.1      1/1        fastEthernet0/0
11.1.1.1/32      I2-E-i  192.168.1.105    115/10     fastEthernet0/0
11.11.11.12/32   I2-I-i  192.168.1.105    115/10     fastEthernet0/0
22.2.0.0/16      I2-I-i  92.168.1.105     115/10     fastEthernet0/0
34.0.0.0/8       I2-E-i  192.168.1.105    115/10     fastEthernet0/0
172.20.32.0/24   Static  192.168.1.1      1/1        fastEthernet0/0
174.20.32.0/24   I2-I-i  192.168.1.105    115/20     fastEthernet0/0
176.20.32.0/24   Connect 176.20.32.1      0/1        loopback1
```

```
    192.168.1.0/24   Connect 192.168.1.214    0/1        fastEthernet0/0
    201.1.1.0/24     I2-E-i  192.168.1.105    115/10     fastEthernet0/0
    201.2.1.0/24     I2-E-i  192.168.1.105    115/10     fastEthernet0/0
    201.3.1.0/24     I2-E-i  192.168.1.105    115/10     fastEthernet0/0
    202.1.1.1/32     I2-E-i  192.168.1.105    115/10     fastEthernet0/0
    207.1.1.0/24     I2-E-i  192.168.1.105    115/10     fastEthernet0/0
```

- Example 5—Indirect Next Hop (* displayed)

```
host1#show ip route
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VR/VRF, *- indirect next-hop

  Prefix/Length     Type      Next Hop      Dst/Met             Intf
------------------ ------- --------------- ---------- ------------------------
21.21.21.2/32      Static  0.0.0.0         1/0        loopback0[V:pe2]
2.2.2.2/32         O-I     30.30.30.2      110/3      ATM2/0.30
                           31.31.31.2      110/3      ATM2/0.31
10.10.10.0/24      Connect 10.10.10.1      0/0        ATM2/0.10
20.20.20.0/24      Connect 20.20.20.1      0/0        ATM2/0.21
4.4.4.4/32         Bgp     2.2.2.2*        200/2
                           3.3.3.3*        200/2
5.5.5.5/32         Bgp     4.4.4.4*        20/2
```

- Example 6—Indirect Next Hop with detail

```
host1#show ip route 4.4.4.4 detail
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF

4.4.4.4/32   Type: Bgp   Distance: 200   Metric: 0   Tag: 0
   Indirect NHop: virtual-router: pe1
   Address 1.1.1.1 Type Bgp Index 1
      NHop: 10.10.10.2 IfIndx: 28 Intf: ATM2/0.10
      NHop: 20.20.20.2 IfIndx: 28 Intf: ATM2/0.20

   Indirect NHop: virtual-router: pe1
   Address 2.2.2.2 Type Bgp Index 2
      NHop: 10.10.10.2 IfIndx: 28 Intf: ATM2/0.10
      NHop: 20.20.20.2 IfIndx: 28 Intf: ATM2/0.20
```

show ip route slot

- Use to display the interface and next hop for an IP address in the routing table of a line module.
- A next hop is displayed only for protocols where ARP is used to resolve the addresses, such as for fastEthernet, gigabitEthernet, bridged Ethernet over ATM, and so on.
- Field descriptions
  - IP address—Address reachable via the interface
  - Interface—Interface type and specifier associated with the IP address; displays "Local Interface" if a special interface index is present in the routing table for special IP addresses, such as broadcast addresses
  - Next Hop—IP address of the next hop router to reach the IP address; displays "*---*" if no next hop is associated with the IP address; displays "Down" if the ECMP set for a specific route on a slot is down
- Example 1

```
host1#show ip route slot 6 10.10.0.231
IP address     Interface            Next Hop
------------   -----------------    ------------
10.10.0.231    fastEthernet 6/0     10.10.0.231
```

- Example 2

```
host1#show ip route slot 9 90.248.1.2
IP address     Interface            Next Hop
------------   -----------------    ------------
90.248.1.2     serial9/23:2         ---
```

- Example 3

```
host1#show ip route slot 9 90.249.255.255
IP address     Interface            Next Hop
------------   -----------------    ------------
90.249.255.255 Local Interface      ---
```

show ip socket statistics

- Use to display basic information about BSD sockets that have been instantiated in the VR in whose context you issue the command. The information includes the connection information (source and destination IP address and port numbers), socket type, the options in effect on the socket, and the socket's state.
- Use the **detailed** keyword to display blocks of extensive information about every socket, such as how many times various APIs have been called and the socket event log. The **detailed** keyword displays information about only the sockets that are associated with the VR in whose context you issue the command or sockets that are not associated with any VR.
- Baselining is not supported for this command.

- Field descriptions
  - *socketNumber ipAddress:portNumber --> ipAddress:portNumber*—Socket and the IP address and port number for each end of the connection, with the E-series router shown on the left and the remote peer on the right
  - type—Type of connection: SOCK_STREAM (uses TCP) or DGRAM (datagram; uses UDP)
  - opts—Options set on the individual sockets
    - SO_DEBUG—Turn on debugging; has no effect
    - SO_ACCEPTCONN—Socket can accept incoming connections
    - SO_REUSEADDR—Allow reuse of the local address
    - SO_KEEPALIVE—Do keepalives on the connection
    - SO_DONTROUTE—Do not route packets, use interface addresses
    - SO_BROADCAST—Broadcasts can be sent over the socket
    - SO_USELOOPBACK—Bypass the hardware if/when possible
    - SO_LINGER—Linger on a close() if data is present
    - SO_OOBINLINE—Leave received out-of-band data in-line
    - SO_REUSEPORT—Allow reuse of local port
  - so_state—State of each socket; knowledge of BSD Sockets API is useful to understand this information
    - SS_NOFDREF—No file table reference any more
    - SS_ISCONNECTED—Socket is connected to a peer
    - SS_ISCONNECTING—Socket is in process of connecting to peer
    - SS_ISDISCONNECTING—Socket is in process of disconnecting
    - SS_CANTSENDMORE—Socket cannot send more data to peer
    - SS_CANTRCVMORE—Socket cannot receive more data from peer
    - SS_RCVATMARK—Socket at mark on input
    - SS_PRIV—Socket is privileged for broadcast, raw
    - SS_NBIO—Socket allows nonblocking operations
    - SS_ASYNC—Socket allows asynchronized I/O notifications
    - SS_ISCONFIRMING—Socket is deciding to accept connection request
  - pending xmit byte count = 0 recv count—Number of bytes that are pending to be sent (queued up) and received
  - Keep alive idle time—Number of seconds before TCP sends an initial keepalive probe to an idle remote node
  - keep alive poll time—Interval in seconds at which TCP sends keepalive probes to idle remote nodes
  - Additional state flags—State of the following flags in the socket_stats structure: ss_Bound, ss_BindError, ss_ListenOk, ss_ListenError, ss_AcceptOk, ss_AcceptError, ss_RsAcceptOk, ss_RsAcceptError, ss_ConnectOk, ss_ConnectErrors, ss_ConnectToOk, ss_ConnectToError, ss_CalledShutdown, and ss_CalledRsSocreate.

- Counters that show how often the indicated routine has been called: so_SendtoCalls, so_SendMsgCalls, so_SendCalls, so_SockWriteCalls, so_SendErrors, so_SentBytes, so_BsdCloseNotClosed, so_RecvBytes, so_RecvErrors, so_RecvFroms, so_Recvs, so_RecvMsgs, so_Reads
- Socket Event Log (most recent at bottom)—Event log on this socket. Each one shows a call to a particular function within the socket library. Includes a repetition counter that displays only nonzero values.
  - Call to sofree()—Call included because in some circumstances an sofree() call does not result in the socket being destroyed (and memory being returned to the free pool)
  - Call to rsSocket()—Call to create the socket using rsSocket() as opposed to socket()
  - Call to socket()—8-bit value indicating how the call went
  - Call to connect()—8-bit value indicating how the call went
  - Call to listen()—8-bit value indicating how the call went
  - Call to accept()—8-bit value indicating how the call went
  - Call to bind()—8-bit value indicating how the call went
  - Call to connectto()—8-bit value indicating how the call went
  - Call to rsAccept()—8-bit value indicating how the call went
  - Call to sobind()—8-bit value indicating how the call went
  - Call to solisten()—8-bit value indicating how the call went
  - Call to soclose()—8-bit value indicating how the call went
  - Call to soabort()—8-bit value indicating how the call went
  - Call to soaccept()—8-bit value indicating how the call went
  - Call to soconnect()—8-bit value indicating how the call went
  - Call to soconnect2()—8-bit value indicating how the call went
  - Call to sodisconnect()—8-bit value indicating how the call went
  - Call to soshutdown()—8-bit value indicating how the call went
  - Call to sowakeup()—8-bit value indicating what kind of wakeup it is. 1 (SELREAD) indicates that data is available on the socket for the application. 2 (SELWRITE) means that more buffer space is available and the application can queue up more data to be transmitted.
  - Call to soclose()—8-bit value indicating how the call went
  - Call to sendto()—16-bit value indicating the return status
  - Call to write()—16-bit value indicating the return status
  - Call to sendmsg()—16-bit value indicating the return status
  - Call to send()—16-bit value indicating the return status
  - Call to recvfrom()—16-bit value indicating the return status
  - Call to recv()—16-bit value indicating the return status
  - Call to recvmsg()—16-bit value indicating the return status
  - Call to read()—16-bit value indicating the return status

- Example 1

```
host1#show ip socket statistics
   5 10.13.5.70:23 --> 10.10.132.71:2000
    type: 1 (SOCK_STREAM)
    opts = 13 SO_DEBUG SO_REUSEADDR SO_KEEPALIVE
    so_state = 177 SS_NOFDREF SS_CANTSENDMORE SS_CANTRCVMORE SS_PRIV

  18 0.0.0.0:23 --> 0.0.0.0:0
    type: 1 (SOCK_STREAM)
    opts = 7 SO_DEBUG SO_ACCEPTCONN SO_REUSEADDR
    so_state = 128 SS_PRIV
```

- Example 2—Additional fields displayed by **detailed** keyword

```
host1#show ip socket statistics detailed
18 0.0.0.0:23 --> 0.0.0.0:0
    type: 1 (SOCK_STREAM)
    opts = 7 SO_DEBUG SO_ACCEPTCONN SO_REUSEADDR
    so_state = 128 SS_PRIV
  pending xmit byte count = 0  recv count 0
  Keep alive idle time = 14400  keep alive poll time = 150
  Additional state flags:
      so_Bound
      so_ListenOk
      ss_CalledRsSocreate

        so_SendtoCalls = 0
       so_SendMsgCalls = 0
          so_SendCalls = 0
     so_SockWriteCalls = 0
         so_SendErrors = 0
          so_SentBytes = 0
  so_BsdCloseNotClosed = 0
          so_RecvBytes = 0
         so_RecvErrors = 0
          so_RecvFroms = 0
              so_Recvs = 0
            so_RecvMsgs = 0
               so_Reads = 0
  Socket Event Log (most recent at bottom)
    rssocket
    sobind - 0
    bind - 0
    solisten - 0
    listen - 0
```

show ip static

- Use to display the status of static routes in the routing table.
- You can specify an IP mask that filters specific routes.
- Field descriptions
  - Prefix—IP address prefix
  - Length—Prefix length
  - Next Hop—IP address of the next hop
  - Met—Number of hops
  - Dist—Administrative distance of the route; see Table 6

- Tag—Tag value of the route
- Intf—Interface type and interface specifier
- Verify—Status of the RTR or BFD operation associated with the specified static route; this field is blank if the **verify** (BFD) or **verify rtr** (RTR) keywords were not specified as part of the **ip route** command. The display can include the following:
  - BFD up/down—Current status of the associated BFD operation
  - operation number—Number of the associated RTR operation
  - up/down—Current status of the associated RTR operation
  - (lr)—Indicates that although the associated RTR operation is currently down, the router will install this route in the routing table, provided that no other static route to the same network prefix is available; this field appears for an RTR operation that is down when the **last-resort** keyword is specified as part of the **ip route verify rtr** command

- Example

```
host1#show ip static
Prefix/Length      Next Hop    Met   Dist   Tag    Intf               Verify
1.1.1.2/32         1.1.1.2       0     1        0 FastEthernet4/0    2 up
1.1.1.2/32         1.1.1.2       0     1        0 FastEthernet4/1
10.10.133.17/32    10.6.128.1    1     1        0 unresolved         1 down
11.11.11.11/32     3.3.3.3       0     1        0 unresolved         1 down(lr)
```

show tcp ack-rst-and-syn

- Use to display the status of TCP ACK, RST, and SYN protection.
- Example

```
host1#show tcp ack-rst-and-syn
TCP Ack Rst and Syn Protection is ENABLED
```

show tcp resequence-buffers

- Use to display the configuration, current per-VR, and per-router state of the TCP resequencing buffer management functions.
- Use the *vrfName* variable to specify a specific VRF for which you want to view information.
- Field descriptions

  TCP Resequence Buffer Management Configuration

  - Global Maximum—Number of buffers that can be on the reordering queues of all connections in all virtual routers
  - Default Per-VR Maximum—Default maximum number of buffers for all connections in a single VR
  - Default Connection Maximum—Default maximum number of buffers for each connection in each virtual router
  - This VR Maximum—Maximum number of outstanding resequencing buffers in the current VR
  - This VR Connection Maximum—Maximum number of outstanding resequencing buffers on any one connection in this VR

TCP Resequence Buffer Management State

- Global buffers in use—Total number of outstanding resequencing buffers in the router
  - High Water—Largest number of outstanding resequencing buffers that the router has experienced since the last reset
- VR Buffers in use—Number of outstanding resequencing buffers in the current virtual router
  - High Water—Largest number of outstanding resequencing buffers for the current virtual router since the last reset
- Buffers Discarded Because Global Limit Exceeded—Number of resequencing buffers discarded because the global limit was reached
- Buffers Discarded Because VR Limit Exceeded—Number of resequencing buffers that have been discarded in this virtual router because the virtual router buffer limit was reached

- Example

```
host1#show tcp resequence-buffers
TCP Resequence Buffer Management Configuration
  Global Maximum: ###
    Default Per-VR Maximum: 250
    Default Connection Maximum: 15
  This VR Maximum: 300
  This VR Connection Maximum: 15

TCP Resequence Buffer Management State
  Global buffers in use: 5
             High Water: 15
      VR Buffers in use: 17
             High Water: 32
  Buffers Discarded Because Global Limit Exceeded: 25
      Buffers Discarded Because VR Limit Exceeded: 15
```

show tcp path-mtu-discovery

- Use to display PMTU information.
- Field descriptions
  - TCP PMTU Discovery—State of the PMTUD functions (ENABLED or DISABLED)
  - Administrative Minimum MTU—Administrative minimum PMTU that is supported or *none* if there is no minimum
  - Administrative Maximum MTU—Administrative maximum PMTU that is supported or none if there is no maximum
  - Timer 1—Value of timer 1 in minutes
  - Timer 2—Value of timer 2 in minutes

  - Black Hole Detect Threshold—Number of retransmissions allowed before TCP/PMTUD assumes that there is a black hole and attempts to reduce impact in the MSS
  - # ICMP TooBigs—Number of ICMP Too Big messages that have been received
  - # ICMP TooBigs for unk. connections—Number of ICMP Too Big messages that have been received which were not for a valid connection
- Example

```
host1#show tcp path-mtu-discovery
TCP PMTU Discovery is ENABLED
    Administrative Minimum MTU: 512
    Administrative Maximum MTU: 65535
    Timer 1: 10 minutes
    Timer 2: 2 minutes
Black Hole Detect Threshold: 0 retransmissions
# ICMP TooBigs: 0
# ICMP TooBigs for unk. connections: 0
```

show tcp paws

- Use to display the TCP PAWS status.
- Example

```
host1#show tcp paws
TCP PAWS is disabled
```

show tcp statistics

- Use to display all TCP statistics.
- Baselining is supported for this command.
- Use the **ip** keyword to display only IPv4 statistics.
- Use the **ipv6** keyword to display only IPv6 statistics.
- Use the **brief** keyword to display summary information or the **detailed** keyword to display extensive information.
- Use the **diagnostic** keyword to display diagnostic information collected on the TCP statistics in addition to the detailed information. This command shows information only for the connections that are active within the context of the VR in which you issue the command.
- Field descriptions
  - TCP Global Statistics Connections:
    - attempted—Number of outgoing TCP connections attempted
    - accepted—Number of incoming TCP connections accepted
    - established—Number of TCP connections established
  - TCP Global Statistics Rcvd:
    - total pkts—Total number of packets received
    - in-sequence pkts—Number of packets received in sequence
    - bytes—Number of bytes received

- chksum err pkts—Number of checksum error packets received
- authentication err pkts—Number of authentication error packets received
- bad offset pkts—Number of bad offset packets received
- short pkts—Number of short packets received
- duplicate pkts—Number of duplicate packets received
- out of order pkts—Number of packets received out of order

- TCP Global Statistics Sent:
  - total pkts—Total number of packets sent
  - data pkts—Number of data packets sent
  - bytes—Number of bytes sent
  - retransmitted pkts—Number of packets retransmitted
  - retransmitted bytes—Number of bytes retransmitted
- Global Diagnostic Data Unknown Connection log—Includes the following global statistics:
  - Source address/port – local port—Shows the 32 most recent TCP connection attempts that were rejected, including the remote node's IP address and port, the local port for the connection attempt, and the number of identical attempts that have been received on that port in a row. The reason for rejection is not given. This information may be useful in tracking down DoS attacks.
  - # connection-reqs rejected—Total number of connection attempts that have been rejected
  - # connection-reqs pending—Current number of connection attempts that are pending, awaiting additional data from the peer
  - # sonewconn calls that fail—Number of calls to sonewconn that have failed. This statistic often indicates that either a socket connection limit has been reached or that there was no memory to hold the socket data structures.
- TCP Session Statistics
  - Local addr—Local address of the TCP connection
  - Local port—Local port number of the TCP connection
  - Remote addr—Remote address of the TCP connection
  - Remote port—Remote port number of the TCP connection
  - State—Current state of the TCP connection
  - Authentication—Authentication status of the TCP connection

- TCP Session Statistics Sent:
  - total pkts—Total number of packets sent on the TCP connection
  - data pkts—Number of data packets sent on the TCP connection
  - bytes—Number of bytes sent on the TCP connection
  - retransmitted pkts—Number of packets retransmitted on the TCP connection
  - retransmitted bytes—Number of bytes retransmitted on the TCP connection
- TCP Session Statistics Rcvd:
  - total pkts—Total number of packets received on the TCP connection
  - in-sequence pkts—Number of packets received in sequence on the TCP connection
  - bytes—Number of bytes received on the TCP connection
  - chksum err pkts—Number of checksum error packets received on the TCP connection
  - bad offset pkts—Number of bad offset packets received on the TCP connection
  - short pkts—Number of short packets received on the TCP connection
  - duplicate pkts—Number of duplicate packets received on the TCP connection
  - out of order pkts—Number of packets received out of order on the TCP connection
- Diagnostics: PRU_ Operations counters—Number of calls for each of the indicated PRU_operations within the TCP service API. These are per-connection statistics.
- Wildcard Matches—Number of packets received that matched this TCP connection due to wildcard matching. Matching is expected for listening server connections, such as Telnet, but is not expected for established connections. This is a per-connection statistic.
- Rcv'd Packets after connection closed—Number of packets received on the connection after the connection has been closed (and before the data structure gets removed). This is a per-connection statistic.
- Connect request rejected—Number of times an incoming connection request was not approved. This is a per-connection statistic.
- Connect request approval pending—Number of times that an incoming connection request was held pending, waiting for a subsequent packet. This is a per-connection statistic.
- New soconnect failed—Number of times a SONEWCONN() was tried on a listening connection and failed. This is a per-connection statistic.
- # Write-Wakeups—Number of times a “write wakeup” occurred on the connection. This is a per-connection statistic.
- # Read wakeups—Number of times a “read wakeup” occurred on the connection. This is a per-connection statistic.

- # receives after close—Number of packets received with data after the connection entered the close-wait state. This is a per-connection statistic.
- Retransmit timer—Current value of the retransmit timer
- Persistence timer—Current value of the persistence timer
- Keepalive timer—Current value of the keepalive timer
- 2MSL timer—Current value of the 2MSL (max segment lifetime) timer
- tcpDisconnect()s—Number of times BsdTcp::tcpDisconnect() was called. This is a per-connection statistic.
- keep T/O pre-estab—Number of times the keepalive timer expired before the connection reached the established state. This is a per-connection statistic.
- tcpkeeptimeo_idle—Number of times the keepalive timer popped, but no keepalive was sent because of connection idle-time considerations. This is a per-connection statistic.
- TCP Connection Event Log (most recent at bottom)—Event log for the TCP connection. It shows the last 32 events that occurred on the connection. The most recent event is at the bottom of the list. This is per-connection data.
  - TCPS_ELOG_PRU_ATTACH
  - TCPS_ELOG_PRU_BIND

The following events can be recorded:

Fast Timeout	Did a PRU_CONNECT
2MSL Timeout	Did a PRU_CONNECT2
Retransmit Timeout	Did a PRU_DISCONNECT
Persist Timeout	Did a PRU_ACCEPT
Received FIN packet	Did a PRU_SHUTDOWN
Received SYN packet	Did a PRU_RCVD
Received Retransmission	Did a PRU_SEND
Transmit a FIN packet	Did a PRU_ABORT
Transmit a SYN packet	Did a PRU_SENSE
Retransmit a packet	Did a PRU_RCVOOB
Did a PRU_ATTACH	Did a PRU_SENDOOB
Did a PRU_DETACH	Did a PRU_SOCKADDR
Did a PRU_BIND	Did a PRU_PEERADDR
Did a PRU_LISTEN	The keepalive timer popped. An 8-bit argument that describes how the timer was handled: ■ Ignored because the session was not established (that is, not in the OPEN state) ■ Ignored due to idle-timeout considerations ■ A packet was sent ■ Ignored because the connection did not have the keepalive option set OR the connection was in the process of closing

- RST/SYN-Ack DoS Protection—Specifies when this function is enabled
  - RSTs acked—Number of RSTs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus RSTs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - SYNs acked—Number of SYNs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus SYNs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - Data Insertions rejected—Number of packets received and dropped because they are believed to have been inserted by an attacker

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been rejected if the protections were enabled. Providing this information can help determine whether attacks are occurring.

- PMTUD information—Information regarding path MTU discovery
  - PMTUD—Status of path MTU discovery on the virtual router: enabled or disabled
  - Administrative Minimum MTU—Minimum MTU that is enabled on any connection; a value of “none” indicates that the minimum is zero (0)
  - Administrative Maximum MTU—Maximum MTU that is enabled on any connection; a value of “none” indicates that the maximum is 65535
  - Timer 1—Amount of time the virtual router waits after receiving an ICMP Too Big message before attempting to increase the path MTU
  - Timer 2—Amount of time the virtual router waits after successfully increasing the MTU before attempting to increase it more
  - # ICMP TooBigs—Number of ICMP Too Big messages that the router has received. When PMTU is disabled, this counter does not increase.
  - # ICMP TooBigs for unk. connection—Number of ICMP Too Big messages that the router has received for TCP connections that do not exist. When PMTU is disabled, this counter does not increase.

- PMTU Increase Attempts—Number of attempts the router has made to increase the PMTU
- Black Hole Detect Threshold—Number of successive transmissions that must occur on a connection before that connection treats retransmissions as indications that something is wrong
- Override MSS—MSS that is advertised to peers, overriding the MSS that is derived from the interface MTU. This line does not appear in the output if you do not set the value.

- MTU/MSS information—Information regarding path MTU/MSS
  - PMTU—Status of MTU/MSS on this virtual router: enabled or disabled
  - MSS in effect—MSS currently being used for transmission to the peer. This number changes while various network events occur to cause the router to increase or decrease its estimate of the MSS.
  - Calculated MSS to peer—MSS that path MTU discovery has calculated (if PMTUD is enabled) to the peer
  - MSS received from peer—MSS that the peer received in a TCP MSS option. If no option is received, the value is zero (0).
  - Application set MSS—MSS that an application might have set for the connection
  - Xmit Interface MSS—MSS for the interface used to transmit packets to the peer; calculated as the interface MTU minus the size of the TCP and IP headers.
  - MSS Sent to Peer—MSS that has been advertised to the peer
  - "ICMP DestUn, Frag Req'd and DF Set" messages—Number of ICMP "Destination Unreachable: Fragmentation Required and DF set" messages that the router has received
  - Number of attempts to increase PMTU—Number of times the router has attempted to increase the PMTU by probing with a packet that is larger than the known MTU
  - Time to next increase attempt—Amount of time, in seconds, until the router retries to increase the MTU
  - Black Hole Detection State—State of the black hole detection mechanism: none, detecting, probable, or unknown
- Out-of-Order Packet Queue Information—Information regarding packet queue buffers
  - Buffers Outstanding—Number of buffers currently on the connection reordering queue
  - High Water—Most buffers that have ever been on the connection reordering queue
  - Buffers discarded—Number of buffers that were discarded because keeping them would have exceeded the connection maximum
- TCP PAWS is [enabled/disabled]—Status of the TCP PAWS option; enabled indicates that PAWS is functioning normally (default mode) for TCP segments; disabled indicates that PAWS is disabled for TCP segments

- Example 1

```
host1#show ip tcp statistics
TCP Global Statistics:
  Connections: 7358 attempted, 4 accepted, 7362 established
        0 dropped, 14718 closed
  Rcvd: 75923 total pkts, 53608 in-sequence pkts, 3120303 bytes
        0 chksum err pkts, 0 authentication err pkts, 0 bad offset pkts
        0 short pkts, 0 duplicate pkts, 0 out of order pkts
  Sent: 82352 total pkts, 44404 data pkts, 657095 bytes
        34 retransmitted pkts, 487 retransmitted bytes

TCP Session Statistics:
  Local addr: 0.0.0.0, Local port: 23
  Remote addr: 0.0.0.0, Remote port: 0
  State: LISTEN Authentication: None
  Rcvd: 4 total pkts, 0 in-sequence pkts, 0 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 0 total pkts, 0 data pkts, 0 bytes
        0 retransmitted pkts, 0 retransmitted bytes

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data pkts, 2304 bytes
        0 retransmitted pkts, 0 retransmitted bytes
 Local addr: 192.168.1.250, Local port: 23
  Remote addr: 192.168.1.139, Remote port: 1038
  State: ESTABLISHED Authentication: None
  Rcvd: 295 total pkts, 159 in-sequence pkts, 299 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 281 total pkts, 210 data pkts, 3089 bytes
        0 retransmitted pkts, 0 retransmitted bytes
```

- Example 2—Additional fields displayed by **diagnostic** keyword

```
host1#show ip tcp statistics diagnostic
...
Global Diagnostic Data
    Unknown Connection log
Source address/port -> local port
         128.127.126.125/124 -> 8080  count: 3
         111.111.111.111/222 -> 3333  count: 4
# connection-reqs rejected: 0
    # connection-reqs pending: 0
    # sonewconn calls that fail: 0
...
```

```
  Diagnostics:
    PRU_ Operations counters:
      PRU_ATTACH: 0
      PRU_DETACH: 0
      PRU_BIND: 1
      PRU_LISTEN: 1
      PRU_CONNECT: 0
      PRU_ACCEPT: 0
      PRU_DISCONNECT: 0
      PRU_SHUTDOWN: 0
      PRU_RCVD: 0
      PRU_SEND: 0
      PRU_ABORT: 0
      PRU_CONTROL: 0
      PRU_SENSE: 0
      PRU_RCVOOB: 0
      PRU_SENDOOB: 0
      PRU_SOCKADDR: 0
      PRU_PEERADDR: 0
      PRU_CONNECT2: 0
      PRU_FASTTIMO: 0
      PRU_SLOWTIMO: 0
      PRU_PROTORCV: 0
      PRU_PROTOSEND: 0
    Wildcard Matches: 2
    Rcv'd Packets after connection closed: 0
    Connect request rejected: 0
    Connect request approval pending 0
    New soconnect failed 0
    # Write-Wakeups: 0
    # Read wakeups 0
    # receives after close 0
    Retransmit timer: 0
    Persistance timer: 0
    Keepalive timer: 0
    2MSL timer: 0
    tcpDisconnect()s: 0
    keep T/O pre-estab: 0
    tcpkeeptimeo_idle: 0
...
TCP Connection Event Log (most recent at bottom)
      TCPS_ELOG_PRU_ATTACH
      TCPS_ELOG_PRU_BIND
```

- Example 3—Additional fields displayed by **detailed** keyword

```
host1#show ip tcp statistics detailed
...

RST/SYN-Ack Protection is: ENABLED
    RSTs acked: 0
    ...Bogus RSTs: 0
    SYNs acked: 0
    ...Bogus SYNs: 0
    Data Insertions rejected: 0
PMTUD Information:      PMTUD: ENABLED
    Administrative Minimum MTU: 512
    Administrative Maximum MTU: none
    Timer 1: 10 minutes
    Timer 2: 2 minutes
```

```
    # ICMP TooBigs: 0
    # ICMP TooBigs for unk. connection: 0
    PMTU Increase Attempts: 17
    Black Hole Detect Threshold: 50 retransmissions
...
MTU/MSS Information
    ENABLED on this connection
    MSS in effect: 536
    Calculated MSS to peer: 536
    MSS received from peer: 0
    Application set MSS: 0
    Xmit Interface MSS: 0
    MSS Sent to Peer: 0
    "ICMP DestUn, Frag Req'd and DF Set" messages: 0
    Number of attempts to increase PMTU: 0
    Time to next increase attempt: 0 seconds
    Black Hole Detection State: none
...
Out-of-order Packet Queue Information

  Buffers Outstanding: 25
           High Water: 28
    Buffers discarded: 15
...
TCP-Paws is disabled
```

show ip traffic

- Use to display statistics about IP traffic.
- You can use the ipTraffic log to show consumable IP traffic to the SRP module; the traffic is filterable per router and IP interface. You can show ICMP, TCP, and UDP traffic with the icmpTraffic, udpTraffic, and tcpTraffic logs.
- Field descriptions
  - IP Statistics Rcvd:
    - router Id—Router ID number
    - total—Number of frames received
    - local destination—Frames with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IP Statistics Frags:
    - reassembled—Number of reassembled packets
    - reasm timed out—Number of reassembled packets that timed out
    - reasm req—Number of requests for reassembly
    - reasm fails—Number of reassembly failures

- frag ok—Number of fragmented packets reassembled successfully
- frag fail—Number of fragmented packets reassembled unsuccessfully
- frag creates—Number of packets created by fragmentation

- IP Statistics Sent:
  - forwarded—Number of packets forwarded
  - generated—Number of packets generated
  - out disc—Number of outbound packets discarded
  - no routes—Number of packets that could not be routed
  - routing discards—Number of packets that could not be routed and were discarded
- IP Statistics Route:
  - routes in table—Number of routes in the routing table
- ICMP Statistics Rcvd:
  - total—Total number of ICMP packets received
  - errors—Number of error packets received
  - dst unreach—Number of packets received with destination unreachable
  - time exceed—Number of packets received with time-to-live exceeded
  - param probs—Number of packets received with parameter errors
  - src quench—Number of source quench packets received
  - redirects—Number of receive packet redirects
  - echo req—Number of echo request (ping) packets
  - echo rpy—Number of echo replies received
  - timestamp req—Number of requests for a timestamp
  - timestamp rpy—Number of replies to timestamp requests
  - addr mask req—Number of mask requests received
  - addr mask rpy—Number of mask replies received
- ICMP Statistics Sent:
  - total—Total number of ICMP packets sent
  - errors—Number of error packets sent
  - dest unreach—Number of packets sent with destination unreachable
  - time excd—Number of packets sent with time-to-live exceeded
  - param prob—Number of packets sent with parameter errors
  - src quench—Number of source quench packets sent
  - redirects—Number of send packet redirects
  - echo req—Number of echo request (ping) packets
  - echo rpy—Number of echo replies sent
  - timestamp req—Number of requests for a timestamp

- timestamp rpy—Number of replies to timestamp requests
- addr mask req—Number of address mask requests sent
- addr mask rpy—Number of replies to address mask requests

- UDP Statistics Rcvd:
  - total—Total number of UDP packets received
  - checksum—Number of checksum error packets received
  - no port—Number of packets received for which no E-series router application listener was listening on the destination port
- UDP Statistics Sent:
  - total—Total number of UDP packets sent
  - errors—Number of error packets sent
- TCP Global Statistics Connections:
  - attempted—Number of outgoing TCP connections attempted
  - accepted—Number of incoming TCP connections accepted
  - established—Number of TCP connections established
  - dropped—Number of TCP connections dropped
  - closed—Number of TCP connections closed
  - currently established—Number of TCP connections currently established
- TCP Global Statistics Rcvd:
  - total pkts—Total number of TCP packets received
  - in-sequence pkts—Number of packets received in sequence
  - bytes—Number of bytes received
  - chksum err pkts—Number of checksum error packets received
  - authentication err pkts—Number of authentication error packets received
  - bad offset pkts—Number of packets received with bad offsets
  - short pkts—Number of short packets received
  - duplicate pkts—Number of duplicate packets received
  - out of order pkts—Number of packets received out of order
- TCP Global Statistics Sent:
  - total pkts—Total number of TCP packets sent
  - data pkts—Number of data packets sent
  - bytes—Number of bytes sent
  - retransmitted pkts—Number of packets retransmitted
  - retransmitted bytes—Number of retransmitted bytes

- OSPF Statistics—Provides statistics on OSPF
- IGMP Statistics—Provides statistics about queries, reports sent or received
- ARP Statistics—Not supported for this version of the router

- Example

```
host1#show ip traffic
IP statistics: Router Id: 172.31.192.217
  Rcvd:  97833 total, 171059 local destination
  0 hdr errors, 0 addr errors
         167 unkn proto, 0 discards
   Frags: 4 reassembled, 30 reasm timed out, 8 reasm req
         0 reasm fails, 145 frag ok, 0 frag fail
         290 frag creates
 Sent:  15 forwarded, 25144 generated, 0 out disc
         0 no routes,0 routing discards
  Route: 57680 routes in table
    0 timestamp req, 0 timestamp rpy
        0 addr mask req, 0 addr mask rpy
ICMP statistics:
  Rcvd: 561 total, 0 errors, 15 dst unreach
        0 time exceed, 0 param probs, 0 src quench
        0 redirects, 0 echo req, 0 echo rpy
        0 timestamp req, 0 timestamp rpy
        0 addr mask req, 0 addr mask rpy
  Sent: 463866 total, 0 errors, 163676 dest unreach
        0 time excd, 0 param prob, 0 src quench
        20 redirects, 463846 echo req, 0 echo rpy
        0 timestamp req, 0 timestamp rpy
        0 addr mask req, 0 addr mask rpy
UDP Statistics:
  Rcvd: 93326 total, 0 checksum errors, 90610 no port
  Sent: 0 total, 0 errors
TCP Global Statistics:
  Connections: 7358 attempted, 4 accepted, 7362 established
        0 dropped, 14718 closed
Rcvd: 75889 total pkts, 53591 in-sequence pkts, 3120283 bytes
        0 chksum err pkts, 0 authentication err pkts, 0 bad offset
        0 short pkts, 0 duplicate pkts, 0 out of order pkts
  Sent: 82318 total pkts, 44381 data pkts, 656321 bytes
        34 retransmitted pkts, 487 retransmitted bytes
OSPF Statistics:
IGMP Statistics:
ARP Statistics:
```

show ip udp statistics

- Use to display UDP statistics.
- Field descriptions
  - UDP Statistics Rcvd:
    - total—Total number of UDP packets received
    - checksum—Number of checksum error packets received
    - no port—Number of packets received for which no E-series router application listener was listening on the destination port
  - UDP Statistics Sent:
    - total—Total number of UDP packets sent
    - errors—Number of error packets sent
- Example

```
host1#show ip udp statistics
UDP Statistics:
  Rcvd: 39196 total, 0 checksum errors, 29996 no port
  Sent: 210 total, 0 errors
```

show profile brief

- Use to list all profile names.
- Field descriptions
  - Profile—Profile names
- Example

```
host1#show profile brief
Profile :
foo
trill
profile4
```

show route-map

- Use to display the configured route maps.
- The displayed information includes the instances of each access list such as **match** and **set** commands.
- Example

```
host1(config)#route-map westford permit 10
host1(config-route-map)#match community 44
host1(config-route-map)#set local-pref 400
host1(config-route-map)#exit
host1(config)#exit
host1#show route-map westford
route-map 1, permit, sequence 10
  Match clauses:
    match community 44
  Set clauses:
    set local-pref 400
```

Chapter 2

Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6) routing on your E-series router; it contains the following sections:

Overview

Internet Protocol version 6 (IPv6) is designed to eventually supersede IP version 4 (IPv4). The intent of this design change is not to take a radical step away from IPv4, but to enhance IP addressing and maintain other IPv4 functions that work well.

The differences between IPv4 and IPv6 include the following:

- Expanded addressing capabilities

  IPv6 increases the size of the IP address from 32 bits to 128 bits. This increased size provides a larger address space and a much larger number of addressable nodes.

- Simplified header format

  Reducing some common processing costs associated with packet handling and streamlining the bandwidth cost of the larger IPv6 header, some IPv4-specific header fields either no longer exist or are now optional in the IPv6 header.

- Traffic flow labelling capabilities

  The ability to label packets for specific traffic flows exists in the IPv6 packet. These labels allow for nondefault quality of service (QoS) or the possibility of "real-time" services.

- Authentication capabilities

  Authentication provides the ability to use extensions for some authentication and data integrity applications.

IPv6 continues to provide the basic packet delivery service for all TCP/IP networks. As a *connectionless* protocol, IPv6 does not exchange control information to establish an end-to-end connection before transmitting data. Instead, just like its IPv4 predecessor, IPv6 continues to rely on protocols in other layers to establish the connection if connection-oriented services are required and to provide error detection and error recovery.

In addition to supporting a revised header structure and an expanded addressing format, the E-series router supports the following IPv6 features:

- Static routes
- ICMPv6
- Ping
- Traceroute
- Routing policy (See *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy* for details.)
- IPv6 B-RAS (See the *JUNOSe Broadband Access Configuration Guide* for details.)
- IPv6 tunnel routing tables

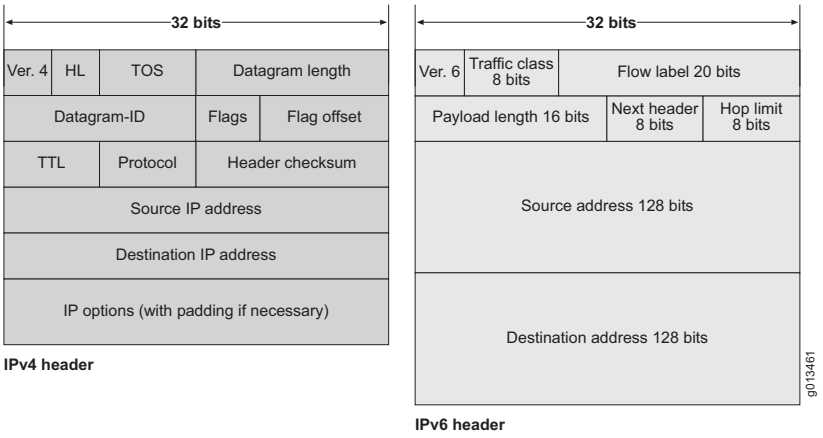
IPv6 Packet Headers

An IPv6 packet is a block of data that contains a header and a payload. The header is the information necessary to deliver the packet to a destination address; the payload is the data that you want to deliver. IPv6 packets can use a standard or an extended format.

IPv4 and IPv6 Header Differences

The main difference between IPv4 and IPv6 resides in their headers. Figure 13 provides a comparison between the two protocol versions.

Figure 13: IPv4 and IPv6 Header Comparison



Standard IPv6 Headers

IPv6 packet headers contain many of the fields found in IPv4 packet headers; some of these fields differ from IPv4. (See Figure 13.)

The 40-byte IPv6 header consists of the following eight fields:

- Version—Indicates the version of the Internet Protocol.
- Traffic class—Previously the type-of-service (ToS) field in IPv4, the traffic class field defines the class-of-service (CoS) priority of the packet. However, the semantics for this field (for example, DiffServ code points) are identical to IPv4.
- Flow label—The flow label identifies all packets belonging to a specific flow (that is, packet flows requiring a specific class of service [CoS]); routers can identify these packets and handle them in a similar fashion.
- Payload length—Previously the total length field in IPv4, the payload length field specifies the length of the IPv6 payload.
- Next header—Previously the protocol field in IPv4, the Next Header field indicates the next extension header to examine.

- Hop limit—Previously the time-to-live (TTL) field in IPv4, the hop limit indicates the maximum number of hops allowed.
- Source address—Identifies the address of the source node sending the packet.
- Destination address—Identifies the final destination node address for the packet.

Extension Headers

In IPv6, extension headers are used to encode optional Internet-layer information. Extension headers are placed between the IPv6 header and the upper-layer header in a packet.

IPv6 enables you to chain extension headers together by using the next header field. The next header field, located in the IPv6 header, indicates to the router which extension header to expect next. If there are no more extension headers, the next header field indicates the upper-layer header (TCP header, UDP header, ICMPv6 header, an encapsulated IP packet, or other items).

IPv6 Addressing

IPv6 increases the size of the IP address from the 32 bits found in IPv4 to 128 bits. This increased size provides for a broader range of addressing hierarchies and a much larger number of addressable nodes.

In addition to the increased size, IPv6 addresses can be of different scopes that categorize what types of applications are suitable for the address. IPv6 does not support broadcast addresses, but uses multicast addresses to serve this role. In addition, IPv6 also defines a new type of address called *anycast*.

Address Representation

IPv6 addresses consist of eight hexadecimal groups. Each hexadecimal group, separated by a colon (:), consists of a 16-bit hexadecimal value. The following is an example of the IPv6 format:

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

A group of *xxxx* represents the 16-bit hexadecimal value. Each individual *x* represents a 4-bit hexadecimal value. The following is an example of a possible IPv6 address:

4FDE:0000:0000:0002:0022:F376:FF3B:AB3F

NOTE: Hexadecimal letters in IPv6 addresses are not case sensitive.

IPv6 Address Compression

IPv6 addresses often contain consecutive hexadecimal fields of zeros. To simplify address entry, you can use two colons (::) to represent the consecutive fields of zeros when typing the IPv6 address. Table 9 provides compressed IPv6 address format examples.

Table 9: Compressed IPv6 Formats

IPv6 Address Type	Full Format	Compressed Format
Unicast	10FB:0:0:0:C:ABC:1F0C:44DA	10FB::C:ABC:1F0C:44DA
Multicast	FD01:0:0:0:0:0:0:1F	FD01::1F
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

NOTE: You can use two colons (::) only once in an IPv6 address to represent hexadecimal fields of consecutive zeros.

IPv6 Address Prefix

An IPv6 address prefix is a combination of an IPv6 prefix (address) and a prefix length. The prefix takes the form *ipv6-prefix/prefix-length* and represents a block of address space (or a network). The *ipv6-prefix* variable follows general IPv6 addressing rules (see RFC 2373 for details). The */prefix-length* variable is a decimal value that indicates the number of contiguous, higher-order bits of the address that make up the network portion of the address. For example, 10FA:6604:8136:6502::/64 is a possible IPv6 prefix.

Address Types

IPv6 can use several types of addresses:

- Unicast—Used to identify a single interface, this release of the E-series router product supports the following unicast address types:
  - Global aggregatable—Provides for aggregation of routing prefixes to limit the number of global routing table entries
  - Link-local—Eliminates the need for a globally unique prefix. Local-link addresses allow communications between devices on a local link.
  - Site-local—Used as private addresses to restrict communication to a domain portion.

NOTE: IPv6 routers must not forward packets that have site-local source or destination addresses outside the site.

  - IPv4-compatible—Contains a standard IPv4 address in the lower-order 32 bits of the address and zeros in the higher-order 96 bits of the address. For example, the format of an IPv4-compatible IPv6 address is 0:0:0:0:0:0:A.B.C.D (or condensed as ::A.B.C.D). In other words, devices using IPv6 use the entire 128-bit IPv4-compatible IPv6 address, whereas IPv4 devices use the IPv4 address embedded within the lower-order 32-bits of the address. You would use IPv4-compatible IPv6 addresses for devices that must support both IPv4 and IPv6 protocols.

- Multicast—Used for sending packets to multiple destinations. A multicast transmission sends packets to all interfaces that are part of a multicast group. The group is represented by the IPv6 destination address of the packet.
- Anycast – Used for a set of interfaces on different nodes. An anycast transmission sends packets to only one of the interfaces associated with the address, not to all of the interfaces. This interface is typically the closest interface, as defined by the routing protocol.
- Loopback—Used by a node to send an IPv6 packet to itself. An IPv6 loopback address functions the same as an IPv4 loopback address.
- Unspecified—Indicates the absence of an IPv6 address. For example, newly initialized IPv6 nodes may use the unspecified address as the source address in their packets until they receive an IPv6 address.

NOTE: IPv6 does not use broadcast addresses; instead, IPv6 uses multicast addresses.

Address Scope

Some unicast and multicast IPv6 addresses contain a value known as *scope*. This value identifies the application suitable for the address.

Unicast addresses support two types of scope—global and local. In addition, there are two types of local scope—link-local addresses and site-local addresses.

Link-local unicast addresses, identified by the first ten bits of the prefix, function within a single network link. You cannot use link-local addresses outside a network link.

Site-local unicast addresses function within a site or an intranet. A site consists of multiple network links, and site-local addresses identify nodes inside the intranet. You cannot use site-local addresses outside the site.

Multicast addresses support 16 different types of scope, including node, link, site, organization, and global scope. A four-bit field in the prefix identifies the scope.

Address Structure

Unicast addresses identify a single interface. The address consists of *n* bits for the prefix and 128-*n* bits for the interface ID.

Multicast addresses identify a set of interfaces. The address is made up of the first 8 bits of all ones, a 4-bit flag field, a 4-bit scope field, and a 112-bit group ID.

11111111 | *flgs* | *scop* | *group ID*

The first octet of ones identifies the address as a multicast address. The flags field identifies whether the multicast address is a well-known address or whether it is a transient multicast address. The scope field identifies the scope of the multicast address. The 112-bit group ID identifies the multicast group.

Similar to multicast addresses, anycast addresses identify a set of interfaces. However, packets are sent to only one of the interfaces, not to all interfaces. Anycast addresses are allocated from the normal unicast address space and cannot be distinguished from a unicast address in format. Therefore, each member of an anycast group must be configured to recognize certain addresses as anycast addresses.

ICMP Support

Internet Control Message Protocol (ICMP) provides a mechanism that enables a router or destination host to report an error in data traffic processing to the original source of the packet. For this release, the E-series router supports ICMP for use in the IPv6 **ping** and **traceroute** commands.

The **ping** and **traceroute** commands help you determine destination reachability within a network.

- Use the **ping ipv6** command to send an ICMP echo request packet. In the following example, the request packet is sent to address 1::1 with a data size of 200 and a timeout value of 10 seconds:

  host1#**ping ipv6 1::1 data-size 200 timeout 10**

- Use the **traceroute ipv6** command to discover routes that router packets follow when traveling to their destination. In the following example, the trace destination address is 1::1, the maximum number of hops of the trace is 20, and the timeout value is 10 seconds:

  host1#**traceroute ipv6 1::1 hop-limit 20 timeout 10**

IPv6 Tunnel Routing Table

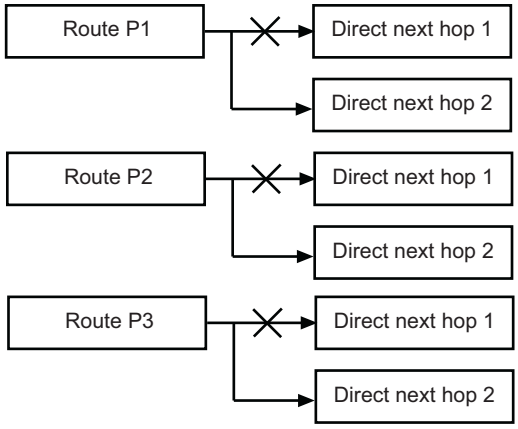
The IPv6 tunnel routing tables include IPv6 routes that point only to tunnels, such as MPLS tunnels. The tunnel routing table is not used for forwarding. Instead, protocols resolve next hops by looking up the routes that point to tunnels. The routes in the tunnel routing table cannot be redistributed. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS* for more information.

Indirect Next Hop Support

The router uses indirect next hops to promote faster network convergence (for example, in BGP networks) by decreasing the number of routing table changes required when a change in the network topology occurs.

Direct next-hops point routes in the routing table toward individual, direct next-hop connections. (See Figure 14.)

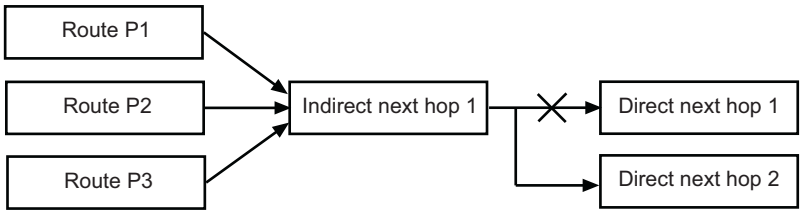
Figure 14: Direct Next Hops



Indirect next hops enable multiple routes in the routing table to point to a single next hop, thereby accelerating convergence. (See Figure 15.)

NOTE: Indirect next hops are not limited to any number of levels. In other words, an indirect next hop can point to a direct next hop or another indirect next hop.

Figure 15: Indirect Next Hops



By using indirect next hops, if a topology change occurs in the network, only the indirect next hop is modified in the routing table, decreasing the number of state changes required to achieve convergence.

Platform Considerations

For information about modules that support IPv6 and Neighbor Discovery on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IP.

For information about modules that support IP on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support IP.

References

For more information about IPv6, consult the following resources:

- RFC 2373—IP Version 6 Addressing Architecture (July 1998)
- RFC 2460—Internet Protocol, Version 6 (IPv6) (December 1998)
- RFC 2461—Neighbor Discovery for IP Version 6 (IPv6) (December 1998)
- RFC 2462—IPv6 Stateless Address Autoconfiguration (December 1998)
- RFC 2463—Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (December 1998)
- RFC 2464—Transmission of IPv6 Packets over Ethernet Networks (December 1998)
- RFC 2465—Management Information Base for IP Version 6: Textual Conventions and General Group (December 1998)
- RFC 2466—Management Information Base for IP Version 6: ICMPv6 Group (December 1998)

You can access these and other Internet RFCs and drafts at the following URL:

http://www.ietf.org

Before You Configure IPv6

Before you configure IPv6, you must create the lower-layer interfaces over which IPv6 traffic flows. In this release, the following modules support IPv6 configuration:

- ATM OC3/STM-1
- ATM OC12/STM-4
- Fast Ethernet (FE-8)
- Gigabit Ethernet (GE)
- 10-Gigabit Ethernet (10GE)
- OC48 POS (PPP only)

For example, to configure an ATM interface:

```
host1(config)#interface atm 1/0
host1(config-if)#atm sonet stm-1
host1(config-if)#no loopback
host1(config-if)#atm clock internal chassis
host1(config-if)#interface atm 1/0.10
host1(config-if)#atm pvc 10 0 20 aal5snap
```

See *JUNOSe Link Layer Configuration Guide, Chapter 1, Configuring ATM* for information about configuring an ATM interface. See *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces* for information about configuring an Ethernet interface.

Configuring an IPv6 License

You must configure an IPv6 license before you can use any IPv6 commands on the E-series router.

license ipv6

- Use to specify an IPv6 license.
- Purchase an IPv6 license to allow IPv6 configuration on the E-series router.

NOTE: Acquire the license from Juniper Networks Customer Services and Support or your Juniper Networks sales representative.

- Example

  host1(config)#**license ipv6** *license-value*

- Use the **no** version to disable the license.

Creating an IPv6 Profile

You can configure an IPv6 interface dynamically by creating a profile. A profile is a set of characteristics that acts as a pattern that can be dynamically assigned to an IPv6 interface. You can manage a large number of IPv6 interfaces efficiently by creating a profile with a specific set of characteristics. In addition, you can create a profile to assign an IPv6 interface to a virtual router.

A profile can contain one or more of the following characteristics:

- address—Configures an IPv6 address on an interface
- mld—Configures the MLD interface
- mtu—Configures the MTU for a network
- nd—Configures Neighbor Discovery (ND) router advertisement characteristics
- policy—Attaches (or removes) a policy to (or from) an interface
- sa-validate—Enables source address validation
- unnumbered—Configures IPv6 on this interface without a specific address
- virtual-router—Specifies a virtual router to which interfaces created by this profile will be attached

NOTE: You can also configure any of these IPv6 characteristics outside the profile configuration mode.

Use the **profile** command from Global Configuration mode to create or edit a profile. See *JUNOSe Link Layer Configuration Guide, Chapter 13, Configuring Dynamic Interfaces Using Bulk Configuration* for information about creating profiles and on other characteristics that can be applied to the profile.

```
host1(config)#profile boston
host1(config-profile)#ipv6 virtual-router warf
host1(config-profile)#ipv6 unnumbered atm 3/0
```

ipv6 address

- Use to add an IPv6 address to an interface or a subinterface.
- Example

```
host1(config)#interface atm 1/0.25
host1(config-if)#ipv6 address 1::1/64
```

NOTE: You can use this command in Interface Configuration or Subinterface Configuration mode.

- Use the **no** version of this command to remove an IPv6 address.

ipv6 nd

- Use to enable the IPv6 Neighbor Discovery process on an interface.
- You can include the following commands in IPv6 profiles to configure Neighbor Discovery route advertisement characteristics. For additional information, see *Using IPv6 Profiles and RADIUS to Configure Neighbor Discovery Route Advertisements* in *Chapter 3, Configuring Neighbor Discovery*.

Command	Description
ipv6 nd	Enables Neighbor Discovery on an interface
ipv6 nd managed-config-flag	Sets the "managed address configuration" flag in IPv6 router advertisements
ipv6 nd other-config-flag	Sets the "other stateful configuration" flag in IPv6 router advertisements
ipv6 nd prefix-advertisement	Specifies IPv6 prefix included in IPv6 router advertisements
ipv6 nd ra-interval	Configures the interval between IPv6 router advertisements
ipv6 nd ra-lifetime	Configures the router advertisement lifetime
ipv6 nd reachable-time	Configures the amount of time the router can reach an IPv6 node after a reachability confirmation event occurs
ipv6 nd suppress-ra	Disables router advertisement transmissions

- Example

  host1(config)#**profile ProfileIpv6South22**
  host1(config-profile)#**ipv6 nd**

- Use the **no** version to disable the Neighbor Discovery process for the profile.

ipv6 mtu

- Use to set the MTU size of IPv6 packets sent on an interface.
- The range is 128–10240.
- Example

  host1(config-if)#**ipv6 mtu 1000**

- Use the **no** version to restore the default MTU size.

ipv6 unnumbered

- Use to set up an unnumbered interface.
- An unnumbered interface does not have an IPv6 address assigned to it. Unnumbered interfaces are often used in point-to-point connections where an IPv6 address is not required.
- This command enables IPv6 processing on an interface without your having to assign an explicit IPv6 address to the interface.

- You supply an interface location that is the type and number of another interface on which the router has an assigned IPv6 address. This interface cannot be another unnumbered interface.
- Example

  host1(config-if)#**ipv6 unnumbered loopback 0**

- Use the **no** version to disable IPv6 processing on an interface.

ipv6 virtual-router

- Use to assign a virtual router to a profile.
- You can configure a virtual router using RADIUS instead of adding one to the profile by using the **ipv6 virtual-router** command.
- Example

  host1(config-profile)#**ipv6 virtual-router VR6**

- Use the **no** version to remove the virtual router assignment.

Assigning a Profile

To assign a profile to an interface, use the **profile** command from Interface mode.

profile

- Use to assign a profile to a PPP interface. The profile configuration is used to dynamically create an upper IP interface.
- Example

  host1(config-if)#**interface atm 3/1.50**
  host1(config-if)#**encapsulation ppp**
  host1(config-if)#**profile boston**

- Use the **no** version to remove the assignment from the interface.

Enabling Source Address Validation

Source address validation verifies that a packet has been sent from a valid source address. When a packet arrives on an interface, the router performs a routing table lookup using the source address. The result from the routing table lookup is an interface to which packets destined for that address are routed. This interface must match the interface on which the packet arrived. If it does not match, the router drops the packet.

CAUTION: When the routing table lookup for a source address contains an ECMP route, the router returns a list of interfaces for multiple next-hops. One of the interfaces in this list must match the interface on which the packet arrived or the router drops the packet. If the ECMP route uses indirect next-hops, the returned list of interfaces does not include interfaces that are reachable by those indirect next-hops. For example, if a packet arrives on an interface with source address validation enabled, and the interface is represented only by an indirect next-hop, a match for that interface does not appear in the list of interfaces from the routing table lookup. The router drops the packet.

ipv6 sa-validate

- Use to enable source address validation. Source address validation verifies that a packet has been sent from a valid source address.
- Example

  host1(config-if)#**ipv6 sa-validate**

- Use the **no** version to disable source address validation.

Establishing a Static Route

You can set a destination to receive and send traffic by a specific route through the network.

ipv6 route

- Use to establish a static IPv6 route.
- You can set a destination to receive and send traffic from and to a network or to use a specific route through the network.
- Example

  host1(config)#**ipv6 route 7fff::0/16 1::1**

- Use the **no** version of this command to remove a static route from the routing table.

Specifying an IPv6 Hop Count Limit

You can specify the maximum number of hops that the router can use in router advertisements and all IPv6 packets.

ipv6 hop-limit

- Use to set the maximum number of hops that the router can use in router advertisements and all IPv6 packets.
- Example

  host1(config)#**ipv6 hop-limit 50**

- Use the **no** version to set the hop limit for IPv6 packets to 255 hops and router advertisements to zero (0) hops (or "unspecified").

Managing IPv6 Interfaces

You can manage IPv6 interfaces in the following ways:

- Disable or reenable an IPv6 interface.

  host1(config-if)#**no ipv6 enable**
  host1(config-if)#**ipv6 enable**

- Set a baseline for IPv6 interface counters.

  host1#**clear ipv6 interface atm 2/0**

- Determine reachability within a network.

  host1#**ping ipv6 1::1**
  host1#**traceroute ipv6 1::1**

clear ipv6 interface

- Use to set a baseline for counters on a specified IPv6 interface.
- Example

  host1#**clear ipv6 interface atm 2/0**

- There is no **no** version.

ipv6 enable

- Use to enable or disable an IPv6 interface at any time.

NOTE: By default, an IPv6 interface is enabled when you first create it.

- Example

  host1(config-if)#**ipv6 enable**

- Use the **no** version of this command to disable IPv6 on an interface or a subinterface.

ping ipv6

- Use to send an ICMP echo request packet to the IPv6 address that you specify.
- Use the **source interface** keywords to specify a source interface other than the one from which the probe originates.
- Use the **source address** keywords to specify a source IP address other than the one from which the probe originates.
- You can specify the following options:
  - **packetCount**—Number of packets to send to the destination IPv6 address. If you specify a zero (0), echo requests packets are sent indefinitely.
  - **data-pattern**—Sets the type of bits contained in the packet to all ones, all zeros, a random mixture of ones and zeros, or a specific hexadecimal data pattern that can range from 0x0–0xFFFFFFFF. The default is all zeros.
  - **data-size**—Sets the number of bytes comprising the IPv6 packet and reflected in the IPv6 header in the range 0–64000; the default is 100 bytes
  - **extended** header attributes—Set the interface type and specifier of a destination address on the router that is configured for external loopback; the command succeeds only if the specified interface is configured for external loopback
  - **sweep-interval**—Specifies the change in the size of subsequent ping packets while sweeping across a range of sizes. For example, you can configure the sweep interval to sweep across the range of packets from 100 bytes to 1000 bytes in increments equal to the sweep interval. By default the router increments packets by one byte; for example, it sends 100, 101, 102, 103, ... 1000. If the sweep interval is 5, the router sends 100, 105, 110, 115, ... 1000.
  - **sweep-sizes**—Enables you to vary the sizes of the echo packets being sent. This capability is useful for determining the minimum sizes of the MTUs configured on the nodes along the path to the destination address. This reduces packet fragmentation, which contributes to performance problems. The default is not to sweep (all packets are the same size).
  - **timeout**—Sets the number of seconds to wait for an ICMP echo reply packet before the connection attempt times out
  - **hop-limit**—Sets the time-to-live hop count in the range 1–255; the default is 255

- The following characters can appear in the display after you issue the **ping** command:
  - !—Reply received
  - .—Timed out while waiting for a reply
  - ?—Unknown packet type
  - A—Admin unreachable
  - b—Packet too big
  - H—Host unreachable
  - N—Network unreachable
  - P—Port unreachable
  - p—Parameter problem
  - S—Source beyond scope
  - t—Hop limit expired (TTL expired)
- Example

  host1#**ping ipv6 1::1**

- There is no **no** version.

traceroute ipv6

- Use to discover the routes that router packets follow when traveling to their destination.
- You can specify:
  - Destination IPv6 address
  - Source interface for each of the transmitted packets
  - Source IPv6 address for each of the transmitted packets
  - Maximum number of hops of the trace and a timeout value
  - Size of the IPv6 packets (not the ICMP payload) in the range 0–64000 bytes sent with the **traceroute** command. Including a size might help locate any MTU problems that exist between your router and a particular device.
  - Hop count in the range 1–255; the default is 32
- You can also force transmission of the packets on a specified interface regardless of what the IPv6 address lookup indicates.
- Example

  host1#**traceroute ipv6 1::1 timeout 10**

- There is no **no** version.

Configuring Shared IPv6 Interfaces

You can create multiple *shared* IPv6 interfaces over the same layer 2 logical interface—for example, atm 5/3.101—enabling more than one IPv6 interface to share the same logical resources.

For additional information about shared interfaces, see *Shared IP Interfaces* on page 55.

To share IPv6 interfaces:

1. Create a layer 2 interface.

   host1(config)#**interface atm 5/3**
   host1(config-if)#**interface atm 5/3.101**

2. (Optional) Create a primary IPv6 interface.

   host1(config-if)#**ipv6 address 1::1/64**
   host1(config-if)#**exit**

3. Create the shared IPv6 interface.

   host1(config)#**interface ipv6 si0**

4. Associate the shared IPv6 interface with the layer 2 interface by the following method:

   host1(config-if)#**ipv6 share-interface atm 5/3.101**

5. To fully configure the shared interface, assign an address (or make the interface unnumbered).

   host1(config-if)#**ipv6 address 1::1/64**

interface ipv6

- Use to create an IPv6 interface for interface sharing.
- Use the specified name to refer to the shared IPv6 interface; you cannot use the layer 2 interface to refer to them, because the shared interface can be moved.
- Example

  host1(config)#**interface ipv6 si1**

- Use the **no** version to delete the IPv6 interface.

ipv6 share-interface

- Use to specify the layer 2 interface used by a shared IPv6 interface. The command fails if the layer 2 interface does not yet exist. The command is not supported (that is, it fails) if you use an RSVP tunnel (for example, **tunnel mpls:1**) to identify the layer 2 interface.
- After creating the shared IPv6 interface, you can configure it as you do any other IPv6 interface.
- The shared interface is operationally up when the layer 2 interface is operationally up.
- You can create operational shared IPv6 interfaces in the absence of a primary IPv6 interface.
- Example

  host1(config-if)#**ipv6 share-interface atm 5/3.101**

- Use the **no** version to remove the association between the layer 2 interface and the shared IPv6 interface. You can delete shared and primary IPv6 interfaces independently.

Adding a Description

The router enables you to add a text description or an alias to an IPv6 interface or subinterface. Adding a description helps you identify the interface and keep track of interface connections.

ipv6 description

- Use to assign a text description or an alias to an IPv6 interface or subinterface.
- The description or alias can be a maximum of 256 characters.
- Use the **show ipv6 interface** command to display the text description.
- Example 1

  host1(config-if)#**ipv6 description boston01 ipv6 interface**

- Example 2

  host1(config-subif)#**ipv6 description dallas05 ipv6 subinterface**

- Use the **no** version to remove the text description or alias.

IPv6 TCP Configuration

IPv6 supports TCP configuration. You use the same commands to configure TCP on IPv6 as you do to configure TCP on IPv4.

Setting MSS for TCP Connections

MSS is used by TCP to define the maximum amount of data that a TCP interface can accept in any single packet (or segment size). The MSS value is typically negotiated during connection establishment and is not renegotiated.

By default, the router uses an MSS value of 1280 bytes and the advertised MSS is derived from the MTU of the transmitting interface. However, you can use the **tcp mss** command to set the MSS for TCP use.

tcp mss

- Use to specify the MSS value for TCP to use.

NOTE: The MSS value is equal to the MTU value minus the IPv6 and TCP headers, so the MSS value is generally 60 bytes less than the MTU.

- Use the *vrfName* variable to specify a VRF to which you want to assign the TCP MSS value.
- Example

  host1(config)#**tcp mss 1000**

- Use the **no** version to remove the MSS value so that the router uses the advertised MSS derived from the MTU of the output interface.

Configuring Path MTU Discovery

IPv6 hosts transmit large amounts of data to other hosts using a series of IPv6 datagrams. To best use resources, increase performance, and avoid difficult reassembly, hosts try to send datagrams that are as large as possible without requiring fragmentation anywhere along the path from the source to the destination. This datagram size is referred to as the *path MTU* (*PMTU*), and it is equal to the smallest MTU for each hop in the path.

Path MTU discovery is the process of discovering the PMTU value and using that value when transmitting IP datagrams.

Enabling PMTU Discovery

Use the **tcp path-mtu-discovery** command to enable PMTU discovery on the active virtual router.

tcp path-mtu-discovery

- Use to enable and configure path MTU discovery on the virtual router.
- Issue the command without any keywords to enable path MTU discovery.
- Issue the **age-timer** keyword to set the time (*minutes*) that TCP waits before attempting to increase the path MTU after receiving an ICMP Too Big message or after previously increasing the PMTU successfully (*minutes2*). The range of these two timers is 1–30 minutes. The timer defaults to 10 minutes.
- Issue the **age-timer indefinite** keyword to disable PMTU aging functions.
- Example 1—Enables path MTU discovery

  host1:VR1(config)#**tcp path-mtu-discovery**

- Example 2—Sets path MTU discovery age timers differently

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 20 15**

- Example 3—Sets path MTU discovery age timers to the same value (5 minutes)

  host1:VR1(config)#**tcp path-mtu-discovery age-timer 5**

- Example 4—Disables path MTU discovery age timers

  host1:VR1(config)#**tcp path-mtu-discovery age-timer indefinite**

- Use the **no** version with a keyword to return the values to their defaults.
- Issue the **no** version without any keywords to disable path MTU discovery on the virtual router.

Limiting PMTU

You can limit calculated PMTU values within a range by using the **tcp path-mtu-discovery max-mtu** and **tcp path-mtu-discovery min-mtu** commands. When specifying PMTU limits, keep the following in mind:

- If a PMTU discovery value is lower than the configured minimum MTU setting, PMTU discovery is disabled for that connection.
- If a PMTU discovery value is larger than the configured maximum MTU setting, the configured maximum MTU setting is used.
- The maximum MTU setting must be greater than the minimum MTU setting.

tcp path-mtu-discovery max-mtu

- Use to limit the maximum MTU size used for the path MTU.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery max-mtu 512**

- Use the **no** version to remove any limitation so that the virtual router uses the path MTU discovery value.

tcp path-mtu-discovery min-mtu

- Use to specify the minimum MTU value used for the path MTU. If the discovered PMTU value is less than the minimum setting, path MTU discovery is disabled for this connection.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery min-mtu 255**

- Use the **no** version to remove any limitation so that the virtual router uses the discovered path MTU value.

Specifying Black Hole Thresholds

Some domains might be configured not to generate certain ICMP messages (like an ICMP destination unreachable message) or to filter all ICMP messages. Under these conditions, the source of oversized ICMP packets never learns that it is sending oversized packets. The device continues sending oversized packets that never get through. This behavior is often referred to as a *black hole*.

A black hole threshold is a limit to the number of times a virtual router can retransmit identical sequences of datagrams before the retransmissions are identified as a problem.

tcp path-mtu-discovery black-hole-detect-threshold

- Use to specify the number of permitted retransmissions before the retransmissions are determined to be a problem.
- Example

  host1:VR1(config)#**tcp path-mtu-discovery black-hole-detect-threshold 200**

- Use the **no** version to disable black hole threshold detection.

Protecting Against TCP RST or SYN DoS Attacks

You can use the **tcp ack-rst-and-syn** command to help protect the router from denial of service (DoS) attacks.

Normally, when it receives an RST or SYN message for an existing connection, TCP attempts to shut down the TCP connection. This action is expected under normal conditions, but someone maliciously generating otherwise valid RST or SYN messages can cause problems for network applications and the network as a whole.

When you enable the **tcp ack-rst-and-syn** command, the router challenges any RST or SYN messages that it receives by sending an ACK message back to the expected source of the message. The source reacts in one of the following ways:

- If the source did send the RST or SYN message, it recognizes the ACK message to be spurious and resends another RST or SYN message. The second RST or SYN message causes the router to shut down the connection.
- If the source did not send the RST or SYN message, the source accepts the ACK message as part of an existing connection. As a result, the source does not send another RST or SYN message and the router does not shut down the connection.

NOTE: Enabling this command slightly modifies the way TCP processes RST or SYN messages to ensure that they are genuine.

tcp ack-rst-and-syn

- Use to help protect the router from TCP RST and SYN denial of service attacks.
- Example

  host1(config)#**tcp ack-rst-and-syn**

- Use the **no** version to disable this protection (the default mode).

Preventing TCP PAWS Timestamp DoS Attacks

The TCP Protect Against Wrapped Sequence (PAWS) number option works by including the TCP timestamp option in all TCP headers to help validate the packet sequence number.

Normally, in PAWS packets that have the timestamps option enabled, hosts use an internal timer to compare the value of the timestamp associated with incoming segments against the last valid timestamp the host recorded. If the segment timestamp is larger than the value of the last valid timestamp, and the sequence number is less than the last acknowledgement sent, the host updates its internal timer with the new timestamp and passes the segment on for further processing.

If the host detects a segment timestamp that is smaller than the value of the last valid timestamp or the sequence number is greater than the last acknowledgement sent, the host rejects the segment.

A remote attacker can potentially determine the source and destination ports and IP addresses of both hosts that are engaged in an active connection. With this information, the attacker might be able to inject a specially crafted segment into the connection that contains a fabricated timestamp value. When the host receives this fabricated timestamp, it changes its internal timer value to match. If this timestamp value is larger than subsequent timestamp values from valid incoming segments, the host determines the incoming segments as being too old and discards them. The flow of data between hosts eventually stops, resulting in a denial of service condition.

Use the **tcp paws-disable** command to disable PAWS processing.

NOTE: Disabling PAWS does not disable other processing related to the TCP timestamp option. This means that even though you disable PAWS, a fabricated timestamp that already exists in the network can still pollute the database and result in a successful DoS attack. Enabling PAWS resets the saved timestamp state for all connections in the virtual router and stops any existing attack.

tcp paws-disable

- Use to disable the Protect Against Wrapped Sequence (PAWS) number option in TCP segments.
- You can specify a VRF context for which you want PAWS disabled.
- Example

  host1(config)#**tcp paws-disable**

- Use the **no** version to restore PAWS processing (the default mode).

Protecting Against TCP Out of Order DoS Attacks

You can use the group of **tcp resequence-buffers** commands to help protect the router from TCP out-of-order packet DoS attacks.

TCP guarantees that applications receive data in order. This means that TCP buffers any out-of-order packets it receives until ordered delivery can occur.

To prevent connections from consuming too many resources, TCP limits the amount of data it accepts to the number of data bytes that the receiver is willing to receive and buffer. TCP does not take into account the buffering scheme that the receiver uses. If the receiver uses a fixed-size receive buffer (that is, buffering all packets) regardless of length, a packet that contains only one data byte might consume many data bytes of buffer space, but only one byte of TCP space.

Under these conditions, an attacker can send a large number of 1-byte packets to an E-series router in which each packet is buffered, consuming an entire packet buffer and eventually consuming a large amount of resources.

To defend against this sort of attack, you can set defaults and limits on the number of outstanding buffers on reordering queues. You can configure these defaults and limits on a per-router, per-virtual router, or per-connection within the virtual router basis.

Limiting Buffers per Router

The **tcp resequence-buffers global-maximum** command enables you to limit the number of outstanding buffers on the entire router.

tcp resequence-buffers global-maximum

- Use to specify a router-wide maximum number of buffers that resequencing queues can contain.
- Specify a value of zero (0) to turn off the limit.
- Example

  host1(config)#**tcp resequence-buffers global-maximum**

- Use the **no** version to revert the global maximum buffer value to its default, 1000 buffers.

Limiting Buffers per Virtual Router

The **tcp resequence-buffers vr-maximum** command and **tcp resequence-buffers default-vr-maximum** command allow you to limit the number of outstanding buffers on existing or newly established virtual routers.

tcp resequence-buffers default-vr-maximum

- Use to specify the default buffer limit assigned to all virtual routers when the virtual router is established.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers default-vr-maximum 200**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

tcp resequence-buffers vr-maximum

- Use to define the maximum number of buffers that the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the limit assignment.
- Example

  host1(config)#**tcp resequence-buffers vr-maximum**

- Use the **no** version to revert the virtual router maximum value to its default, 100 buffers.

Limiting Buffers per Connection

The **tcp resequence-buffers connection-maximum** command and **tcp resequence-buffers default-connection-maximum** command allow you to limit the number of outstanding buffers on existing or newly established connections.

tcp resequence-buffers connection-maximum

- Use to define the maximum number of buffers that connections on the current or specified virtual router can use.
- Specify a value of zero (0) to turn off the connection maximum.
- Example

  host1(config)#**tcp resequence-buffers connection-maximum 50**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

tcp resequence-buffers default-connection-maximum

- Use to specify the default buffer limit assigned to all TCP connections on a virtual router unless a specific limit is set for the VR in which the connection is established.
- Specify a value of zero (0) buffers to turn off the default limit.
- Example

  host1(config)#**tcp resequence-buffers default-connection-maximum 100**

- Use the **no** version to revert the connection maximum value to its default, 10 buffers.

Configuring Equal-Cost Multipath Load Sharing

Equal-cost multipath (ECMP) sets are formed when the router finds routing table entries for the same destination with equal cost. The router then balances traffic across these sets of equal-cost paths by using hashed mode.

Hashed Mode

Hashed mode uses hashing of source and destination addresses to determine which of the available paths in the ECMP set to use. Hashed mode is the default ECMP mode of operation.

Defining Maximum Paths

You can add routing table entries manually (as static routes), or they are formed as routers discover their neighbors and exchange routing tables (via OSPF, BGP, and other routing protocols).

The **maximum paths** command controls the maximum number of parallel routes that the routing protocol (BGP, IS-IS, OSPF, or RIP) can support.

maximum-paths

- Use to control the maximum number of parallel routes that the routing protocol supports.
- The maximum number of routes can be in the range 1–16 for BGP, IS-IS, OSPF, or RIP.
- Example

  host1(config-router)#**maximum-paths 2**

- Use the **no** version to restore the default value, 1 for BGP or 4 for IS-IS, OSPF, or RIP.

Fast Reroute Protection

If a link goes down, ECMP uses fast reroute protection to shift packet forwarding to use operational links, thereby decreasing packet loss. Fast reroute protection updates ECMP sets for the interface without having to wait for the route table update process. When the next route table update occurs, a new ECMP set can be added with fewer links or the route might point to a single next hop.

CAUTION: To provide ECMP fast reroute functionality in the event of an interface failure, the members of an equal cost multipath must be resolved to corresponding interfaces. If the member is an indirect next hop, the interface is obtained by using the forwarding equivalence class (FEC) to which the member points. This method of resolving members occurs only if the FEC, pointed to by the indirect next hop, is either an interface or a direct next hop.

An indirect next hop member is not resolved to an interface if it points to another indirect next hop or to an equal cost multipath. ECMP fast reroute functionality is not available if any interfaces that correspond to unresolved indirect next hop members go down.

If you modify an indirect next hop member to point to a different FEC (that is, a different interface, direct next hop, indirect next hop, or ECMP), the indirect next hop member is not resolved for the new changes.

Removing an IPv6 Configuration

To remove an IPv6 configuration from the virtual router, issue the **no ipv6** command.

no ipv6

- Use to remove IPv6 configuration from the virtual router.
- Example

  host1(config)#**no ipv6**

NOTE: The E-series router automatically starts IPv6 processing when you begin configuring an IPv6 interface. However, by issuing the **ipv6** command without using the **no** option, you can create an IPv6 processing instance with no IPv6 configuration.

Clearing IPv6 Routes

To clear dynamic IPv6 routes from the routing table, use the **clear ipv6 routes** command. To clear the routes for a specific IPv6 network, specify the IPv6 prefix. To clear all dynamic IPv6 routes, using the * (asterisk) option.

clear ipv6 routes

- Use to clear IPv6 routes.
- To clear routes in a specific IPv6 network, specify an IPv6 prefix.
- To clear all dynamic IPv6 routes, use the * (asterisk) option.
- Example

  host1(config)#**clear ipv6 routes ***

- There is no **no** version.

Creating Static IPv6 Neighbors

To create static IPv6 neighbors, use the **ipv6 neighbor** command.

ipv6 neighbor

- Use to create static IPv6 neighbors.
- Example

  host1(config)#**ipv6 neighbor 1::10 fastEthernet 1/0 0002.7dfa.0034**

- Use the **no** version of this command to delete the neighbor.

Clearing Dynamic IPv6 Neighbors

To clear dynamic IPv6 neighbors, use the **clear ipv6 neighbor** command. Using the **include-statics** keyword clears both dynamic neighbors and static neighbors. Using the **statics-only** keyword clears only IPv6 static neighbors.

clear ipv6 neighbors

- Use to clear all dynamic IPv6 neighbors.
- Use the **include-statics** keyword to clear both dynamic neighbors and static neighbors. Use the **statics-only** keyword to clear only IPv6 static neighbors.
- Example

  host1(config)#**clear ipv6 neighbors**

- There is no **no** version.

Monitoring IPv6

This section explains how to set an IPv6 statistics baseline and use the **show** commands to view your IPv6 configuration, monitor IPv6 interfaces and statistics, and view IPv6 neighbors. Many of these show commands also contain Neighbor Discovery information.

System Event Logs

To troubleshoot and monitor IPv6, use the following system event logs:

- ipv6General—IPv6 general information
- ipv6Interface—IPv6 interface events
- ipv6ProfileMgr—IPv6 profile manager events
- ipv6RouteTable—IPv6 routing table events
- ipv6Traffic—IPv6 frame transmit and receive events

For more information about using event logs, see the *JUNOSe System Event Logging Reference Guide, Chapter 1,* .

Establishing a Baseline

IPv6 statistics are stored in system counters. The only way to reset the system counters is to reboot the system. You can, however, establish a baseline for IPv6 statistics by setting a group of reference counters to zero (0).

baseline ipv6

- Use to set a baseline for IPv6 statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **udp** keyword to set a baseline for UDP statistics
- Use the **delta** keyword with IPv6 **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ipv6**

- There is no **no** version

baseline ipv6 interface

- Use to set a statistical baseline for a specified IPv6 interface.
- Example

  host1#**baseline ipv6 interface atm 2/0.100**

- There is no **no** version.

baseline tcp

- Use to set a statistics baseline for all (both IPv4 and IPv6) TCP statistics or for only IPv4 or IPv6 statistics.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **ipv6** keyword to implement a baseline for only IPv6 statistics.
- Use the **delta** keyword with IP **show** commands to specify that baselined statistics are to be shown.
- Example 1

  host1#**baseline tcp**

- Example 2

  host1#**baseline ipv6 tcp**

- There is no **no** version.

IPv6 show Commands

You can monitor the following aspects of IPv6 using **show ipv6** commands:

To Display	Command
General IPv6 information	**show ipv6**
IPv6 addresses	**show ipv6 address**
IPv6 forwarding table	**show ipv6 forwarding table slot**
IPv6 Interfaces	**show ipv6 interface**
IPv6 neighbors	**show ipv6 neighbors**
IPv6 profile information	**show ipv6 profile**
Active IPv6 protocol information	**show ipv6 protocols**
IPv6 route redistribution configuration	**show ipv6 redistribute**
IPv6 routes	**show ipv6 route**
IPv6 router advertisements received	**show ipv6 routers**
IPv6 static routes	**show ipv6 static**
IPv6 statistics/traffic	**show ipv6 traffic**
IPv6 UDP information	**show ipv6 udp statistics**
IPv6 license string	**show license ipv6**
IPv6 TCP information	**show tcp statistics** **show ipv6 tcp statistics**

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string that you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

show ipv6

- Use to display general IPv6 information.
- Example

```
host1#show ipv6
 Ipv6 Unicast Routing: Enabled
 Default hop limit: not specified
 Number of interfaces: 2
 Default interface source address/mask: fe80::90:1a00:210:fd0/128
```

show ipv6 address
show ipv6 interface

- Use to display detailed or summary information for a particular IPv6 address or interface or for all interfaces.
- The default for the **show ipv6 interface command** is all interface types and all interfaces.
- Use **brief** or **detail** keywords with the **show ipv6 interface command** to display different levels of information.
- Field descriptions
  - Description—Optional description for the interface or address specified
  - Network Protocols—Network protocols configured on this interface
  - Link local address—Local IPv6 address of this interface
  - Internet address—External address of this interface
  - IPv6 statistics Rcvd:
    - local destination—Frames with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IPv6 statistics Sent:
    - generated—Number of packets generated
    - no routes—Number of packets that could not be routed
    - discards—Number of packets that could not be routed that were discarded
  - ICMPv6 statistics Rcvd:
    - total—Total number of received packets
    - errors—Error packets received
    - destination unreach—Packets received with destination unreachable
    - admin unreach—Packets received because the destination was administratively unreachable (for example, the packet encountered a firewall filter)
    - parameter problem—Packets received with parameter errors
    - time exceeded—Packets received with time-to-live exceeded
    - pkt too big—Number of packet-too-big messages received that indicate a packet was too large to forward because of the allowed MTU size
    - redirects—Received packet redirects
    - echo requests—Echo request (ping) packets
    - echo replies—Echo replies received
    - rtr solicits—Number of received router solicitations

  - rtr advertisements—Number of received router advertisements
  - neighbor solicits—Number of received neighbor solicitations
  - neighbor advertisements—Number or received neighbor advertisements
  - Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests received from within a group to which the interface is assigned
- ICMPv6 statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received
  - destination unreach—Packets received with destination unreachable
  - admin unreach—Packets sent because the destination was administratively unreachable (for example, due to a firewall filter)
  - parameter problem—Packets received with parameter errors
  - time exceeded—Packets received with time-to-live exceeded
  - pkt too big—Number of packet-too-big messages sent because a received packet was too large to forward because of the allowed MTU size
  - redirects—Received packet redirects
  - echo requests—Echo request (ping) packets
  - echo replies—Echo replies received
  - rtr solicits—Number of sent router solicitations
  - rtr advertisements—Number of sent router advertisements
  - neighbor solicits—Number of sent neighbor solicitations
  - neighbor advertisements—Number of sent neighbor advertisements
  - Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests sent to a group of which the interface is assigned
- Operational MTU—Value of the MTU
- Administrative MTU—Value of the MTU if it has been administratively overridden using the configuration
- Operational speed—Speed of the interface
- Administrative speed—Value of the speed if it has been administratively overridden using the configuration
- Creation type—Method by which the interface was created (static or dynamic)
- ND reachable time—Amount of time (in milliseconds) that the neighbor is expected to remain reachable
- ND duplicate address detection attempts—Number of times that the router attempts to determine a duplicate address
- ND neighbor solicitation retransmission interval—Interval in which the router retransmits neighbor solicitations

- ND proxy—Indicates whether the router will reply to solicitations on behalf of a known neighbor
- ND RA source link layer—Indicates whether the RA includes the link layer
- ND RA interval—Interval (in seconds) of the neighbor discovery router advertisement
- ND RA lifetime—Lifetime (in seconds) of the neighbor discovery router advertisement
- ND RA managed flag—State of the neighbor discovery router advertisement managed flag
- ND RA other config flag—State of the neighbor discovery router advertisement other config flag
- ND RA advertising prefixes—Configured advertisement prefixes for neighbor discovery router advertisement
- In Received Packets, Bytes—Total number of packets and bytes received on this interface
  - Unicast Packets, Bytes—Unicast packets and bytes received on the IPv6 interface; link-local received multicast packets (non-multicast-routed frames) are counted as unicast packets
  - Multicast Packets, Bytes—Multicast packets and bytes received on the IPv6 interface which are then multicast-routed are counted as multicast packets
- In Total Dropped Packets, Bytes—Total number of inbound packets and bytes dropped on this interface
  - In Policed Packets—Packets that were received and dropped because of rate limits
  - In Invalid Source Address Packets—Packets received with invalid source address (for example, spoofed packets)
  - In Error Packets—Number of packets received with errors
  - In Discarded Packets—Packets received that were discarded for reasons other than rate limits, errors, and invalid source address
- Out Forwarded Packets, Bytes—Total number of packets and bytes that were sent from this interface
  - Unicast Packets, Bytes—Unicast packets and bytes that were sent from this interface
  - Multicast Routed Packets, Bytes—Multicast packets and bytes that were sent from this interface

- Out Total Dropped Packets—Total number of outbound packets and bytes dropped by this interface
  - Out Scheduler Dropped Packets, Bytes—Number of outbound packets and bytes dropped by the scheduler
  - Out Policed Packets, Bytes—Number of outbound packets and bytes dropped because of rate limits
  - Out Discarded Packets—Number of outbound packets that were discarded for reasons other than those dropped by the scheduler and those dropped because of rate limits
- IPv6 policy—Type (input, output, local-input) and name of policy
  - rate-limit-profile—Name of profile
  - classifier-group entry—Entry index
  - Committed—Number of packets and bytes conforming to the committed access rate
  - Conformed—Number of packets and bytes that exceed the committed access rate but conform to the peak access rate
  - Exceeded—Number of packets and bytes exceeding the peak access rate
- queue, traffic class, bound to ipv6—Queue and traffic class bound to the specified IPv6 interface
  - Queue length—Number of bytes in queue
  - Dropped committed packets, bytes—Total number of committed packets and bytes dropped by this interface
  - Dropped conformed packets, bytes—Total number of conformed packets and bytes dropped by this interface
  - Dropped exceeded packets, bytes—Total number of exceeded packets and bytes dropped by this interface

- Example 1

```
host1#show ipv6 address 5:1:1::2
FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce
  Internet address: 5:1:1::2/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 12, Bytes 1260
    Unicast Packets 5, Bytes 588
    Multicast Packets 7, Bytes 672
```

```
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 21, Bytes 2352
    Unicast Packets 21, Bytes 2352
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 2

```
host1#show ipv6 address detail 5:1:1::2
FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce

  Internet address: 5:1:1::2/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         3 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 5 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  ICMPv6 statistics:
    Rcvd:  12 total, 0 errors
           0 rtr solicits, 7 rtr advertisements
           1 neighbor solicits, 1 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  31 total, 0 errors
           0 rtr solicits, 16 rtr advertisements
           5 neighbor solicits, 5 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 12, Bytes 1260
    Unicast Packets 5, Bytes 588
    Multicast Packets 7, Bytes 672
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 3

```
host1#show ipv6 interface
null0 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44
  Unnumbered Interface: Corresponding Numbered Interface not specified or
removed
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
```

```
  Link local address: fe80::90:1a00:740:31ce
  Internet address: 5:1:1::2/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled

  In Received Packets 13, Bytes 1356
    Unicast Packets 5, Bytes 588
    Multicast Packets 8, Bytes 768
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd
  Internet address: 6:1:1::1/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
```

```
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

loopback5 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44
  Internet address: 10:1:1:0:290:1aff:fe40:1d44/64 (eui-64)
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

    IPv6 policy input ipv6InPol25
    rate-limit-profile Rlp2Mb classifier-group clgA entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp8Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy output ipv6PolOut2
    rate-limit-profile RlpOutA classifier-group clgB entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile RlpOutB
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy local-input ipv6PolLocIn5
    rate-limit-profile Rlp1Mb classifier-group clgC entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
```

```
    rate-limit-profile Rlp5Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 4

```
host1#show ipv6 interface FastEthernet 9/0.6
FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd
  Internet address: 6:1:1::1/64
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

    IPv6 policy input ipv6InPol25
    rate-limit-profile Rlp2Mb classifier-group clgA entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
```

```
    rate-limit-profile Rlp8Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy output ipv6PolOut2
    rate-limit-profile RlpOutA classifier-group clgB entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile RlpOutB
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy local-input ipv6PolLocIn5
    rate-limit-profile Rlp1Mb classifier-group clgC entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp5Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

- Example 5

```
host1#show ipv6 interface detail
null0 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44

  Unnumbered Interface: Corresponding Numbered Interface not specified or
removed
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  Neighbor Discovery is disabled

  ICMPv6 statistics:
    Rcvd:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0

FastEthernet9/1.5 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop5
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31ce

  Internet address: 5:1:1::2/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         3 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 5 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  ICMPv6 statistics:
    Rcvd:  13 total, 0 errors
           0 rtr solicits, 8 rtr advertisements
           1 neighbor solicits, 1 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
```

```
    Sent:  31 total, 0 errors
           0 rtr solicits, 16 rtr advertisements
           5 neighbor solicits, 5 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 13, Bytes 1356
    Unicast Packets 5, Bytes 588
    Multicast Packets 8, Bytes 768
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 22, Bytes 2480
    Unicast Packets 22, Bytes 2480
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 8, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 8

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/1.5
    Queue length 0 bytes
    Forwarded packets 4, bytes 680
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd

  Internet address: 6:1:1::1/64
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies

  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
    ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface
```

```
  ICMPv6 statistics:
    Rcvd:  0 total, 0 errors
           0 rtr solicits, 0 rtr advertisements
           0 neighbor solicits, 0 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects
    Sent:  13 total, 0 errors
           0 rtr solicits, 9 rtr advertisements
           2 neighbor solicits, 2 neighbor advertisements
           Group membership: 0 queries, 0 responses, 0 reductions
           0 redirects

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

loopback5 line protocol IpLoopback is up, ipv6 is up
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:1d44

  Internet address: 10:1:1:0:290:1aff:fe40:1d44/64 (eui-64)
IPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 generated, 0 no routes, 0 discards

ICMPv6 statistics:
  Rcvd:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
  Sent:  0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 echo requests
         0 echo replies
```

```
Operational MTU 1500 Administrative MTU 0
Operational speed 100000000 Administrative speed 0
Creation type Static
Neighbor Discovery is disabled

ICMPv6 statistics:
  Rcvd:  0 total, 0 errors
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
         0 redirects
  Sent:  0 total, 0 errors
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
         0 redirects

In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Total Dropped Packets 0, Bytes 0
  In Policed Packets 0
  In Invalid Source Address Packets 0
  In Error Packets 0
  In Discarded Packets 0

Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Total Dropped Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0
  Out Discarded Packets 0
  IPv6 policy input ipv6InPol25
  rate-limit-profile Rlp2Mb classifier-group clgA entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile Rlp8Mb
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
IPv6 policy output ipv6PolOut2
  rate-limit-profile RlpOutA classifier-group clgB entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile RlpOutB
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
IPv6 policy local-input ipv6PolLocIn5
  rate-limit-profile Rlp1Mb classifier-group clgC entry 1
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
  rate-limit-profile Rlp5Mb
    Committed: 0 packets, 0 bytes
    Conformed: 0 packets, 0 bytes
    Exceeded:  0 packets, 0 bytes
```

```
    queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0
```

- Example 6

```
host1#show ipv6 interface brief
      Interface            IPv6-Address     Status  Protocol    Description
---------------------- -------------------- ------- -------- ------------------
null0                  Unnumbered           up      up
FastEthernet9/1.5      5:1:1::2/64          up      up       IPv6 interface in
                                                             Virtual Router Hop
                                                             5
FastEthernet9/0.6      6:1:1::1/64          up      up       IPv6 interface in
                                                             Virtual Router Hop
                                                             6
loopback5              10:1:1:0:290:1aff:fe up      up
                       40:1d44/64
```

show ipv6 forwarding-table slot

- Use to display details on the forwarding table for a specific line module only when IPv6 is configured on the router. These details include the memory used by each virtual router configured on the line module and free memory available on the module.
- The Load Errors field records any failed routing table distribution attempt as an error. Attempts can fail for many reasons during normal operation; a failed attempt does not necessarily indicate a problem. It is normal to see many Load Errors per day.
- If the Status field does not indicate Valid, then the routing table distribution has failed constantly for that VR. It is normal and appropriate behavior for the Status field to indicate Valid while the Load Error field increases daily.
- Field descriptions
  - Free Memory—Amount of routing table memory free on the line module, in kilobytes
  - Virtual Router—Name of the virtual routers configured on the line module
  - Memory (KB)—Amount of routing table memory consumed by the virtual router, in kilobytes
  - Load Errors—Count of errors made while loading the routing table on the line module
  - Status—Whether the routing table for the virtual router is valid
- Example

```
host1#show ipv6 forwarding-table slot 9
Free Memory = 32766 KB (99.99%)
         Virtual Router            Memory(KB)    Load Errors      Status
------------------------------ -------------- ------------- --------------
default                        -              -             Not Resident
1                              2              0             Valid
```

show ipv6 neighbors

- Use to display IPv6 Neighbor Discovery cache information static entries, dynamic entries, or both.
- Use the **static** keyword to display only static entries
- Use the **dynamic** keyword to display only dynamic entries
- Use the **summary** keyword to display summary information
- Field descriptions
  - Interface—Neighbor interface
  - IPv6-Address—IPv6 address for the interface
  - Type—Type of interface (dynamic, static)
  - Hardware Addr—Layer 2 address of the interface
  - State—State of the interface (delay, incomplete, probe, reachable, stale)
  - Age—Amount of time (in seconds) since the router contacted the neighbor
  - By type—List by neighbor type (global, link-local, anycast, and unknown)
  - By state—List by neighbor state (reachable, incomplete, stale, probe, delay, an init)
  - IPv6 address conflicts—Number of conflicts during or after duplicate address detection resolution
- Example 1

```
host1#show ipv6 neighbors
      Interface            IPv6-Address      Type   Hardware Addr  State  Age
---------------------- --------------------- ------- -------------- ----- -----
FastEthernet4/1        1::1                 dynamic 0090.1a40.05e5 reach 3
```

- Example 2

```
host1#show ipv6 neighbors summary
Total IPv6 neighbors: 7
By type: 5 global, 2 link-local, 0 anycast, 0 unknown
By state: 5 reachable, 0 incomplete, 2 stale, 0 probe, 0 delay, 0 init
IPv6 address conflicts: 0 during DAD resolution, 0 after DAD resolution
```

show ipv6 profile

- Use to display information about a specific IPv6 profile.
- Field descriptions
  - IPv6 profile—Profile name
  - Unnumbered interface—Specifier for the unnumbered interface or none if the interface is numbered
  - Router—Router name

- Access Route Addition—Enabled or disabled
- Source-Address Validation—Enabled or disabled
- Administrative MTU—MTU size

- Example

```
host1#show ipv6 profile foo
IPv6 profile : foo
Unnumbered interface on : loopback 0
Router                     : r1
Access Route Addition      : Enabled
Source-Address Validation  : Disabled
Administrative MTU         : 0
```

show ipv6 protocols

- Use to display configured protocols.
- Field descriptions
  - Local router ID—Router ID of the local router
  - Local AS—AS number of local router
  - Administrative state—Administrative state of the protocol
  - Operational state—Operational state of the protocol
  - Shutdown in overload state—Status of shutdown in an overload state
  - Default local preference—Default value for local preference
  - IGP synchronization—Indicates whether synchronization is enabled or disabled
  - Default originate—Indicates whether network 0.0.0.0 is redistributed into BGP
  - Auto summary—Status of autosummary
  - Always compare MED—Status of always compare MED
  - Compare MED within confederation—Status of compare MED within a confederation
  - Advertise inactive routes—Status of Advertise inactive routes
  - Advertise best external router to internal peers—Status of Advertise best external router to internal peers
  - Enforce first AS—Status of Enforce first AS
  - Missing MED as worst—Status of Missing MED as worst
  - Route flap dampening—Status of route dampening
  - Log neighbor changes—Status of Log neighbor changes
  - Fast External Fallover—Status of Fast External Fallover
  - Maximum received AS-path length—Maximum AS-path length received
  - BGP administrative distances—External, internal, and local BGP administrative distances
  - Client-to-client reflection—Whether client-to-client reflection is configured
  - Cluster ID—Cluster IDs

- Route-target filter—Status of Route-target filter
- Default IPv4-unicast—Status of Default IPv4-unicast
- Local-RIB version—RIB version
- Local-FIB version—FIB version
- Neighbor(s)—BGP neighbors (if configured)
- Networks for which routing is occurring
- Aggregate Generation for Unicast Routes

- Example 1

```
host1#show ipv6 protocols
Routing Protocol is "bgp 100"
  Local router ID 1.1.1.1, local AS 100
  Administrative state is Start
  Operational state is Up
  Shutdown in overload state is disabled
  Default local preference is 100
  IGP synchronization is enabled
  Default originate is disabled
  Auto summary is enabled
  Always compare MED is disabled
  Compare MED within confederation is disabled
  Advertise inactive routes is disabled
  Advertise best external route to internal peers is disabled
  Enforce first AS is disabled
  Missing MED as worst is disabled
  Route flap dampening is disabled
  Log neighbor changes is disabled
  Fast External Fallover is disabled
  No maximum received AS-path length
  BGP administrative distances are 20 (ext), 200 (int), and 200 (local)
  Client-to-client reflection is enabled
  Cluster ID is 1.1.1.1
  Route-target filter is enabled
  Default IPv4-unicast is enabled
  Local-RIB version 8. FIB version 8.
  Neighbor(s):
    No neighbors are configured
  Routing for Networks:
  Aggregate Generation for Unicast Routes:
```

- Example 2

```
host1#show ipv6 protocols summary
bgp 100
```

show ipv6 redistribute

- Use to display configured route redistribution policy.
- Field descriptions
  - To—Protocol that routes are distributed into
  - From—Protocol that routes are distributed from
  - status—Redistribution status
  - route map name—Name of the route map
- Example

```
host1#show ipv6 redistribute
To bgp, From static is enabled with route map foo
To bgp, From connected is enabled without a route map
```

show ipv6 route

- Use to display the current state of the routing table, including routes not used for forwarding.
- You can display all routes, a specific route, detailed information about all or a specific route, or summary counters for the routing table.
- Field descriptions
  - Prefix—IPv6 address prefix
  - Length—Prefix length
- Type—Protocol type (possible route types include: Bgp, Connect, Idrp, Igrp, Invalid, Isis, Ndisc, Ospf, Other, Rip, Static)
  - Dst (or Distance)—Administrative distance for the route
  - Met (or Metric)—Number of hops
  - Intf (or Interface)—Interface type and interface specifier
  - NextHop—The configured next hop address for this interface
  - IfIndex—An autogenerated value for the next hop interface
- Example 1

```
host1#show ipv6 route
         Prefix/Length             Type   Dst/Met         Intf
------------------------------- ------- ------- --------------------
1::/16                          Connect 0/0     loopback1
5::/64                          Connect 0/0     ATM4/0.15
6::/64                          Static  1/0     ATM4/0.15
2003::/16                       Static  1/0     ATM4/0.15
```

- Example 2

```
host1#show ipv6 route summary
6 total routes, 408 bytes in route entries
0 isis routes
0 rip routes
2 static routes
2 connected routes
0 bgp routes
0 ospf routes
2 other internal routes
0 access routes
```

```
0 internally created access host routes

Last route added/deleted: null by Local
At WED JAN 22 2003 09:53:33 UTC
```

- Example 3

```
host1#show ipv6 route 5::/64 detail
5::/64 Type:local  Distance:0  Metric:0
        NextHop: 1::2  IntfIndex 10007 Intf ATM4/0.15
```

show ipv6 routers

- Use to display IPv6 router advertisement information received.
- Use the conflicts keyword to display router advertisements that differ from the advertisements configured
- Field descriptions
  - Route—Router for which this information applies
  - Hops—Number of hops that the router uses in router advertisements
  - Lifetime—Lifetime (in seconds) of the neighbor discovery router advertisement
  - AddrFlag—State of the neighbor discovery router advertisement managed flag
  - OtherFlag—State of the neighbor discovery router advertisement other config flag
  - Reachable time—Amount of time (in milliseconds) that the neighbor is expected to remain reachable
  - Retransmit time—Interval in which the router retransmits neighbor solicitations
  - Prefix—IPv6 network number to include in router advertisements
  - Autoconfig—When present, indicates that local host links use the specified prefix for IPv6 autoconfiguration
  - Valid lifetime—Amount of time in seconds that the router advertises the IPv6 prefix as valid
  - preferred lifetime—Amount of time in seconds that the router advertises the specified IPv6 prefix as preferred
- Example 1

```
host1#show ipv6 routers
Router FE80::83B3:60A4 on FastEthernet2/0, last update 3 min
  Hops 0, Lifetime 6000 sec, AddrFlag=0, OtherFlag=0
  Reachable time 0 msec, Retransmit time 0 msec
  Prefix 3FFE:C00:8007::800:207C:4E37/96 autoconfig
    Valid lifetime -1, preferred lifetime -1

Router FE80::290:27FF:FE8C:B709 on FastEthernet2/1, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0
  Reachable time 0 msec, Retransmit time 0 msec
```

- Example 2

```
host1#show ipv6 routers conflicts
Router FE80::203:FDFF:FE34:7039 on FastEthernet1/0, last update 1 min,
CONFLICT
   Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0
   Reachable time 0 msec, Retransmit time 0 msec
   Prefix 2003::/64 onlink autoconfig
     Valid lifetime -1, preferred lifetime -1
```

show ipv6 static

- Use to display the status of static routes in the routing table.
- You can specify an IP mask that filters specific routes.
- Field descriptions
  - Prefix—IP address prefix
  - Length—Prefix length
  - Next Hop—IP address of the next hop
  - Dst—Administrative distance of the route
  - Met—Number of hops
  - Interface—Interface type and interface specifier
- Example

```
host1#show ipv6 static
     Prefix/Length               NextHop          Dst/Met      Interface
------------------------ ------------------------ ------- --------------------
6::/64                   5::2                     1/0     ATM4/0.15
2003::/16                5::1                     1/0     ATM4/0.15
```

show ipv6 traffic

- Use to display statistics about IPv6 traffic.
- Field descriptions
  - IPv6 statistics Rcvd:
    - total—Total number of packets received
    - local destination—Number of packets received with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IPv6 statistics Sent:
    - forwarded—Number of packets forwarded
    - generated—Number of packets generated
    - out disc—Number of packets that could not be routed that were discarded

- IPv6 statistics Mcast:
  - received—Number of multicast packets received
  - forwarded—Number of multicast packets forwarded
- IPv6 statistics (Routes)—Number of routes currently in the routing table
- ICMPv6 statistics Rcvd:
  - total—Total number of received packets
  - errors—Error packets received
  - destination unreach—Packets received with destination unreachable
  - admin unreach—Packets received because the destination was administratively unreachable (for example, the packet encountered a firewall filter)
  - parameter problem—Packets received with parameter errors
  - time exceeded—Packets received with time-to-live exceeded
  - pkt too big—Number of packet-too-big messages received that indicate a packet was too large to forward because of the allowed MTU size
  - redirects—Received packet redirects
  - echo requests—Echo request (ping) packets
  - echo replies—Echo replies received
  - rtr solicits—Number of received router solicitations
  - rtr advertisements—Number of received router advertisements
  - neighbor solicits—Number of received neighbor solicitations
  - neighbor advertisements—Number of received neighbor advertisements
  - Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests received from within a group to which the interface is assigned
- ICMP statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received
  - destination unreach—Packets received with destination unreachable
  - admin unreach—Packets sent because the destination was administratively unreachable (for example, due to a firewall filter)
  - parameter problem—Packets received with parameter errors
  - time exceeded—Packets received with time-to-live exceeded
  - pkt too big—Number of packet-too-big messages sent because a received packet was too large to forward because of the allowed MTU size
  - redirects—Received packet redirects
  - echo requests—Echo request (ping) packets
  - echo replies—Echo replies received

- rtr solicits—Number of sent router solicitations
- rtr advertisements—Number of sent router advertisements
- neighbor solicits—Number of sent neighbor solicitations
- neighbor advertisements—Number of sent neighbor advertisements
- Group membership (queries, responses, reductions)—Number of queries, responses, and reduction requests sent to a group to which the interface is assigned

- UDP Statistics Rcvd:
  - total—Total number of received packets
  - checksum errors—Checksum error packets received
  - no port—No port error packets received
- UDP Statistics Sent:
  - total—Total number of received packets
  - errors—Error packets received

- Example

```
host1#show ipv6 traffic
IPv6 statistics:
  Rcvd:  0 total, 0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Sent:  0 forwarded, 0 generated
         0 out disc
  Mcast: 0 received 0 forwarded
  Routes: 7 in routing table

ICMPv6 statistics:
  Rcvd:  0 total, 0 errors
         0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 redirects
         0 echo requests, 0 echo replies
         0 rtr solicits, 0 rtr advertisements
         0 neighbor solicits, 0 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions
  Sent:  3 total, 0 errors
         0 destination unreach, 0 admin unreach, 0 parameter problem
         0 time exceeded, 0 pkt too big, 0 redirects
         0 echo requests, 0 echo replies
         0 rtr solicits, 0 rtr advertisements
         2 neighbor solicits, 1 neighbor advertisements
         Group membership: 0 queries, 0 responses, 0 reductions

UDP Statistics:
  Rcvd: 0 total, 0 checksum errors, 0 no port
  Sent: 0 total, 0 errors
```

show ipv6 udp statistics

- Use to display IPv6 UDP statistics.
- Example

```
host1#show ipv6 udp statistics
UDP Statistics:
  Rcvd: 0 total, 0 checksum errors, 0 no port
  Sent: 0 total, 0 errors
```

show license ipv6

- Use to display the IPv6 license key configured on the router.
- Example

```
host1#show license ipv6
Ipv6 license is ipv6_license
```

show tcp statistics

- Use to display all TCP statistics (both IPv4 and IPv6).
- Baselining is supported for this command.
- Use the **ip** keyword to display only IPv4 statistics.
- Use the **ipv6** keyword to display only IPv6 statistics.
- Use the **brief** keyword to display summary information or the **detailed** keyword to display extensive information.
- Use the **diagnostic** keyword to display diagnostic information collected on the TCP statistics in addition to the detailed information. This command shows information only for the connections that are active within the context of the VR in which you issue the command.
- Field descriptions
  - TCP Global Statistics Connections:
    - attempted—Number of outgoing TCP connections attempted
    - accepted—Number of incoming TCP connections accepted
    - established—Number of TCP connections established
  - TCP Global Statistics Rcvd:
    - total pkts—Total number of packets received
    - in-sequence pkts—Number of packets received in sequence
    - bytes—Number of bytes received
    - chksum err pkts—Number of checksum error packets received
    - authentication err pkts—Number of authentication error packets received
    - bad offset pkts—Number of bad offset packets received
    - short pkts—Number of short packets received
    - duplicate pkts—Number of duplicate packets received
    - out of order pkts—Number of packets received out of order

- TCP Global Statistics Sent:
  - total pkts—Total number of packets sent
  - data pkts—Number of data packets sent
  - bytes—Number of bytes sent
  - retransmitted pkts—Number of packets retransmitted
  - retransmitted bytes—Number of bytes retransmitted
- Global Diagnostic Data Unknown Connection log—Includes the following global statistics:
  - Source address/port – local port—Shows the 32 most recent TCP connection attempts that were rejected, including the remote node's IP or IPv6 address and port, the local port for the connection attempt, and the number of identical attempts that have been received on that port in a row. The reason for rejection is not given. This information may be useful in tracking down DoS attacks.
  - # connection-reqs rejected—Total number of connection attempts that have been rejected
  - # connection-reqs pending—Current number of connection attempts that are pending, awaiting additional data from the peer
  - # sonewconn calls that fail—Number of calls to sonewconn that have failed. This statistic often indicates that either a socket connection limit has been reached or that there was no memory to hold the socket data structures.
- TCP Session Statistics
  - Local addr—Local address of the TCP connection
  - Local port—Local port number of the TCP connection
  - Remote addr—Remote address of the TCP connection
  - Remote port—Remote port number of the TCP connection
  - State—Current state of the TCP connection
  - Authentication—Authentication status of the TCP connection
- TCP Session Statistics Sent:
  - total pkts—Total number of packets sent on the TCP connection
  - data pkts—Number of data packets sent on the TCP connection
  - bytes—Number of bytes sent on the TCP connection
  - retransmitted pkts—Number of packets retransmitted on the TCP connection
  - retransmitted bytes—Number of bytes retransmitted on the TCP connection
- TCP Session Statistics Rcvd:
  - total pkts—Total number of packets received on the TCP connection
  - in-sequence pkts—Number of packets received in sequence on the TCP connection
  - bytes—Number of bytes received on the TCP connection

  - chksum err pkts—Number of checksum error packets received on the TCP connection
  - bad offset pkts—Number of bad offset packets received on the TCP connection
  - short pkts—Number of short packets received on the TCP connection
  - duplicate pkts—Number of duplicate packets received on the TCP connection
  - out of order pkts—Number of packets received out of order on the TCP connection
- Diagnostics: PRU_ Operations counters—Number of calls for each of the indicated PRU_operations within the TCP service API. These are per-connection statistics.
- Wildcard Matches—Number of packets received that matched this TCP connection due to wildcard matching. Matching is expected for listening server connections, such as Telnet, but is not expected for established connections. This is a per-connection statistic.
- Rcv'd Packets after connection closed—Number of packets received on the connection after the connection has been closed (and before the data structure gets removed). This is a per-connection statistic.
- Connect request rejected—Number of times an incoming connection request was not approved. This is a per-connection statistic.
- Connect request approval pending—Number of times that an incoming connection request was held pending, waiting for a subsequent packet. This is a per-connection statistic.
- New soconnect failed—Number of times a SONEWCONN() was tried on a listening connection and failed. This is a per-connection statistic.
- # Write-Wakeups—Number of times a “write wakeup” occurred on the connection. This is a per-connection statistic.
- # Read wakeups—Number of times a “read wakeup” occurred on the connection. This is a per-connection statistic.
- # receives after close—Number of packets received with data after the connection entered the close-wait state. This is a per-connection statistic.
- Retransmit timer—Current value of the retransmit timer
- Persistence timer—Current value of the persistence timer
- Keepalive timer—Current value of the keepalive timer
- 2MSL timer—Current value of the 2MSL (max segment lifetime) timer
- tcpDisconnect()s—Number of times BsdTcp::tcpDisconnect() was called. This is a per-connection statistic.
- keep T/O pre-estab—Number of times the keepalive timer expired before the connection reached the established state. This is a per-connection statistic.
- tcpkeeptimeo_idle—Number of times the keepalive timer popped, but no keepalive was sent because of connection idle-time considerations. This is a per-connection statistic.

- TCP Connection Event Log (most recent at bottom)—Event log for the TCP connection. It shows the last 32 events that occurred on the connection. The most recent event is at the bottom of the list. This is per-connection data.
  - TCPS_ELOG_PRU_ATTACH
  - TCPS_ELOG_PRU_BIND

The following events can be recorded:

Fast Timeout	Did a PRU_CONNECT
2MSL Timeout	Did a PRU_CONNECT2
Retransmit Timeout	Did a PRU_DISCONNECT
Persist Timeout	Did a PRU_ACCEPT
Received FIN packet	Did a PRU_SHUTDOWN
Received SYN packet	Did a PRU_RCVD
Received Retransmission	Did a PRU_SEND
Transmit a FIN packet	Did a PRU_ABORT
Transmit a SYN packet	Did a PRU_SENSE
Retransmit a packet	Did a PRU_RCVOOB
Did a PRU_ATTACH	Did a PRU_SENDOOB
Did a PRU_DETACH	Did a PRU_SOCKADDR
Did a PRU_BIND	Did a PRU_PEERADDR
Did a PRU_LISTEN	The keepalive timer popped. An 8-bit argument that describes how the timer was handled: ■ Ignored because the session was not established (that is, not in the OPEN state) ■ Ignored due to idle-timeout considerations ■ A packet was sent ■ Ignored because the connection did not have the keepalive option set OR the connection was in the process of closing

- RST/SYN-Ack DoS Protection—Specifies when this function is enabled
  - RSTs acked—Number of RSTs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus RSTs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - SYNs acked—Number of SYNs received and then acknowledged by the TCP stack.

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been acknowledged if the protections were enabled. Providing this information can help determine whether attacks are occurring.

  - Bogus SYNs—Number of RSTs that were judged to be invalid (that is, their timer expired) and therefore ignored
  - Data Insertions rejected—Number of packets received and dropped because they are believed to have been inserted by an attacker

NOTE: This count is maintained even when the protection functions are disabled. The value indicates the count of packets that would have been rejected if the protections were enabled. Providing this information can help determine whether attacks are occurring.

- PMTUD information—Information regarding path MTU discovery
  - PMTUD—Status of path MTU discovery on the virtual router: enabled or disabled
  - Administrative Minimum MTU—Minimum MTU that is enabled on any connection; a value of “none” indicates that the minimum is zero (0)
  - Administrative Maximum MTU—Maximum MTU that is enabled on any connection; a value of “none” indicates that the maximum is 65535
  - Timer 1—Amount of time the virtual router waits after receiving an ICMP Too Big message before attempting to increase the path MTU
  - Timer 2—Amount of time the virtual router waits after successfully increasing the MTU before attempting to increase it more
  - # ICMP TooBigs—Number of ICMP Too Big messages that the router has received. When PMTU is disabled, this counter does not increase.
  - # ICMP TooBigs for unk. connection—Number of ICMP Too Big messages that the router has received for TCP connections that do not exist. When PMTU is disabled, this counter does not increase.

- PMTU Increase Attempts—Number of attempts the router has made to increase the PMTU
- Black Hole Detect Threshold—Number of successive transmissions that must occur on a connection before that connection treats retransmissions as indications that something is wrong
- Override MSS—MSS that is advertised to peers, overriding the MSS that is derived from the interface MTU. This line does not appear in the output if you do not set the value.

- MTU/MSS information—Information regarding path MTU/MSS
  - PMTU—Status of MTU/MSS on this virtual router: enabled or disabled
  - MSS in effect—MSS currently being used for transmission to the peer. This number changes while various network events occur to cause the router to increase or decrease its estimate of the MSS.
  - Calculated MSS to peer—MSS that path MTU discovery has calculated (if PMTUD is enabled) to the peer
  - MSS received from peer—MSS that the peer received in a TCP MSS option. If no option is received, the value is zero (0).
  - Application set MSS—MSS that an application might have set for the connection
  - Xmit Interface MSS—MSS for the interface used to transmit packets to the peer; calculated as the interface MTU minus the size of the TCP and IP headers.
  - MSS Sent to Peer—MSS that has been advertised to the peer
  - "ICMP DestUn, Frag Req'd and DF Set" messages—Number of ICMP "Destination Unreachable: Fragmentation Required and DF set" messages that the router has received
  - Number of attempts to increase PMTU—Number of times the router has attempted to increase the PMTU by probing with a packet that is larger than the known MTU
  - Time to next increase attempt—Amount of time, in seconds, until the router retries to increase the MTU
  - Black Hole Detection State—State of the black hole detection mechanism: none, detecting, probable, or unknown
- Out-of-Order Packet Queue Information—Information regarding packet queue buffers
  - Buffers Outstanding—Number of buffers currently on the connection reordering queue
  - High Water—Most buffers that have ever been on the connection reordering queue
  - Buffers discarded—Number of buffers that were discarded because keeping them would have exceeded the connection maximum
- TCP PAWS is [enabled/disabled]—Status of the TCP PAWS option; enabled indicates that PAWS is functioning normally (default mode) for TCP segments; disabled indicates that PAWS is disabled for TCP segments

- Example 1

```
host1#show ipv6 tcp statistics
TCP Global Statistics:
  Connections: 7358 attempted, 4 accepted, 7362 established
        0 dropped, 14718 closed
  Rcvd: 75923 total pkts, 53608 in-sequence pkts, 3120303 bytes
        0 chksum err pkts, 0 authentication err pkts, 0 bad offset pkts
        0 short pkts, 0 duplicate pkts, 0 out of order pkts
  Sent: 82352 total pkts, 44404 data pkts, 657095 bytes
        34 retransmitted pkts, 487 retransmitted bytes

TCP Session Statistics:
  Local addr: 0.0.0.0, Local port: 23
  Remote addr: 0.0.0.0, Remote port: 0
  State: LISTEN Authentication: None
  Rcvd: 4 total pkts, 0 in-sequence pkts, 0 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 0 total pkts, 0 data pkts, 0 bytes
        0 retransmitted pkts, 0 retransmitted bytes

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data

  Local addr: 192.168.1.250, Local port: 23
  Remote addr: 10.10.0.77, Remote port: 2170
  State: ESTABLISHED Authentication: None
  Rcvd: 61 total pkts, 34 in-sequence pkts, 41 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 64 total pkts, 45 data pkts, 2304 bytes
        0 retransmitted pkts, 0 retransmitted bytes
 Local addr: 192.168.1.250, Local port: 23
  Remote addr: 192.168.1.139, Remote port: 1038
  State: ESTABLISHED Authentication: None
  Rcvd: 295 total pkts, 159 in-sequence pkts, 299 bytes
        0 chksum err pkts, 0 bad offset pkts, 0 short pkts
        0 duplicate pkts, 0 out of order pkts
  Sent: 281 total pkts, 210 data pkts, 3089 bytes
        0 retransmitted pkts, 0 retransmitted bytes
```

- Example 2—Additional fields displayed by **diagnostic** keyword

```
host1#show tcp statistics diagnostic
...
Global Diagnostic Data
    Unknown Connection log
Source address/port -> local port
         128.127.126.125/124 -> 8080  count: 3
         111.111.111.111/222 -> 3333  count: 4
# connection-reqs rejected: 0
    # connection-reqs pending: 0
    # sonewconn calls that fail: 0
...
```

```
  Diagnostics:
    PRU_ Operations counters:
      PRU_ATTACH: 0
      PRU_DETACH: 0
      PRU_BIND: 1
      PRU_LISTEN: 1
      PRU_CONNECT: 0
      PRU_ACCEPT: 0
      PRU_DISCONNECT: 0
      PRU_SHUTDOWN: 0
      PRU_RCVD: 0
      PRU_SEND: 0
      PRU_ABORT: 0
      PRU_CONTROL: 0
      PRU_SENSE: 0
      PRU_RCVOOB: 0
      PRU_SENDOOB: 0
      PRU_SOCKADDR: 0
      PRU_PEERADDR: 0
      PRU_CONNECT2: 0
      PRU_FASTTIMO: 0
      PRU_SLOWTIMO: 0
      PRU_PROTORCV: 0
      PRU_PROTOSEND: 0
    Wildcard Matches: 2
    Rcv'd Packets after connection closed: 0
    Connect request rejected: 0
    Connect request approval pending 0
    New soconnect failed 0
    # Write-Wakeups: 0
    # Read wakeups 0
    # receives after close 0
    Retransmit timer: 0
    Persistance timer: 0
    Keepalive timer: 0
    2MSL timer: 0
    tcpDisconnect()s: 0
    keep T/O pre-estab: 0
    tcpkeeptimeo_idle: 0
...
TCP Connection Event Log (most recent at bottom)
      TCPS_ELOG_PRU_ATTACH
      TCPS_ELOG_PRU_BIND
```

- Example 3—Additional fields displayed by **detailed** keyword

```
host1#show tcp statistics detailed
...

RST/SYN-Ack Protection is: ENABLED
    RSTs acked: 0
    ...Bogus RSTs: 0
    SYNs acked: 0
    ...Bogus SYNs: 0
    Data Insertions rejected: 0
PMTUD Information:      PMTUD: ENABLED
    Administrative Minimum MTU: 512
    Administrative Maximum MTU: none
    Timer 1: 10 minutes
    Timer 2: 2 minutes
```

```
    # ICMP TooBigs: 0
    # ICMP TooBigs for unk. connection: 0
    PMTU Increase Attempts: 17
    Black Hole Detect Threshold: 50 retransmissions
...
MTU/MSS Information
    ENABLED on this connection
    MSS in effect: 536
    Calculated MSS to peer: 536
    MSS received from peer: 0
    Application set MSS: 0
    Xmit Interface MSS: 0
    MSS Sent to Peer: 0
    "ICMP DestUn, Frag Req'd and DF Set" messages: 0
    Number of attempts to increase PMTU: 0
    Time to next increase attempt: 0 seconds
    Black Hole Detection State: none
...
Out-of-order Packet Queue Information

  Buffers Outstanding: 25
           High Water: 28
    Buffers discarded: 15
...
TCP-Paws is disabled
```

Chapter 3

Configuring Neighbor Discovery

This chapter describes how to configure Neighbor Discovery (ND) on your E-series router; it contains the following sections:

Overview

Though not a true protocol, routers and hosts (nodes) use Neighbor Discovery (ND) messages to determine the link-layer addresses of neighbors that reside on attached links and to overwrite invalid cache entries. Hosts also use ND to find neighboring routers that can forward packets on their behalf.

In addition, nodes use ND to actively track the ability to reach neighbors. When a router (or the path to a router) fails, nodes actively search for alternatives to reach the destination.

IPv6 Neighbor Discovery corresponds to a number of the IPv4 protocols — ARP, ICMP Router Discovery, and ICMP Redirect. However, Neighbor Discovery provides many improvements over the IPv4 set of protocols. These improvements address the following:

- Router discovery—How a host locates routers residing on an attached link.
- Prefix discovery—How a host discovers address prefixes for destinations residing on an attached link. Nodes use prefixes to distinguish between destinations that reside on an attached link and those destinations that it can reach only through a router.
- Parameter discovery—How a node learns various parameters (link parameters or Internet parameters) that it places in outgoing packets.
- Address resolution—How a node uses only a destination IPv6 address to determine a link-layer address for destinations on an attached link.
- Next-hop determination—The algorithm that a node uses for mapping an IPv6 destination address into a neighbor IPv6 address (either the next router hop or the destination itself) to which it plans to send traffic for the destination.
- Neighbor unreachability detection—How a node determines that it can no longer reach a neighbor.
- Duplicate address detection—How a node determines whether an address is already in use by another node.

Platform Considerations

For information about modules that support Neighbor Discovery on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support Neighbor Discovery.

For information about modules that support Neighbor Discovery on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support Neighbor Discovery.

References

For more information about Neighbor Discovery, consult the following resource:

- RFC 2461—Neighbor Discovery for IP Version 6 (IPv6) (December 1998)

You can access these and other Internet RFCs and drafts at the following URL:

http://www.ietf.org

Before You Configure Neighbor Discovery

Before you configure Neighbor Discovery, you must configure IPv6. For information about configuring IPv6, see *Chapter 2, Configuring IPv6*.

Configuring Ethernet interfaces to function with IPv6 requires Neighbor Discovery configuration for the interface.

NOTE: IPv6 Neighbor Discovery is fully supported when configured on broadcast interfaces. IPv6 neighbor discovery supports only router advertisement characteristics when configured on PPP interfaces.

Configuring Neighbor Discovery

To configure Neighbor Discovery:

1. Access an IPv6 interface.

   host1(config)#**interface fastEthernet 3/0**
   host1(config-if)#

2. Configure the current IPv6 interface to send neighbor solicitations and to respond with neighbor advertisements.

   host1(config)#**ipv6 nd**

NOTE: This command is redundant when configuring Neighbor Discovery over Ethernet, because router advertisements are automatically sent on Ethernet interfaces. However, unless explicitly enabled, IPv6 router advertisements are not sent on other types of interfaces.

3. (Optional) Configure the interface to retry sending neighbor solicitations using a specified interval.

   host1(config-if)#**ipv6 nd ns-interval 500**

4. (Optional) Configure the interface to assume that a neighbor is reachable for a specified time after a reachable confirmation event.

   host1(config-if)#**ipv6 nd reachable-time 30000**

5. (Optional) Configure the interface to suppress router advertisements, as well as replies to router solicitations.

   host1(config-if)#**ipv6 nd suppress-ra**

6. (Optional) Configure the interface to suppresses the source link-layer option in IPv6 router advertisement transmissions. This action forces neighbors to solicit the router link layer explicitly, and may prove necessary when enabling inbound load sharing across multiple link-layer addresses.

   host1(config-if)#**ipv6 nd suppress-ra-source-link-layer**

7. (Optional) Configure the interface to send router advertisements at a specified interval.

   host1(config-if)#**ipv6 nd ra-interval 500**

8. (Optional) Configure the router advertisement lifetime in seconds.

   host1(config-if)#**ipv6 nd ra-lifetime 900**

9. (Optional) Configure the router advertisement to list a specified prefix, for a valid lifetime and preferred lifetime. The following example also advertises the prefix as reachable on link and that the router can use it as part of the stateless address configuration.

   host1(config-if)#**ipv6 nd prefix-advertisement 2002:1::/64 60000 45000 onlink autoconfig**

10. (Optional) Configure the router advertisement to contain the “managed address configuration” flag.

    host1(config-if)#**ipv6 nd managed-config-flag**

11. (Optional) Configure the router advertisement to contain the “other stateful configuration” flag.

    host1(config-if)#**ipv6 nd other-config-flag**

12. (Optional) Enable active solicitations.

    host1(config-if)#**ipv6 nd active-solicitations**

Using IPv6 Profiles and RADIUS to Configure Neighbor Discovery Route Advertisements

In addition to the CLI-based configuration of Neighbor Discovery, you can also use IPv6 profiles to configure Neighbor Discovery route advertisements for dynamically configured interfaces. In addition, you can use RADIUS to configure the prefix in Neighbor Discovery route advertisements for dynamically configured interfaces.

When you configure either a profile-based or RADIUS-based Neighbor Discovery router advertisement, the following considerations apply:

- You can advertise one IPv6 prefix per interface.
- The router advertisement must have a prefix length of 64. For the Ipv6-NdRa-Prefix attribute, the prefix length is in the following format, in which 0040 indicates the prefix length of 64.

  0x 0040 xxxx xxxx xxxx xxxx

NOTE: If both an IPv6 profile and RADIUS are configured for Neighbor Discovery router advertisement, the prefix value returned in RADIUS VSA 26-129 takes precedence over the prefix specified in the IPv6 profile configuration.

IPv6 Profile-Based Configuration

The JUNOSe software enables you to use profiles to dynamically configure IPv6 interfaces. When you create an IPv6 profile, you can also include Neighbor Discovery route advertisement characteristics, which are then configured on the dynamically-created IPv6 interfaces.

You can include the following commands in IPv6 profiles to configure Neighbor Discovery route advertisement characteristics.

Command	Description
ipv6 nd	Enables Neighbor Discovery on an interface
ipv6 nd managed-config-flag	Sets the “managed address configuration” flag in IPv6 router advertisements
ipv6 nd other-config-flag	Sets the “other stateful configuration” flag in IPv6 router advertisements
ipv6 nd prefix-advertisement	Specifies which IPv6 prefixes are included in IPv6 router advertisements
ipv6 nd ra-interval	Configures the interval between IPv6 router advertisements
ipv6 nd ra-lifetime	Configures the router advertisement lifetime
ipv6 nd reachable-time	Configures the amount of time the router can reach an IPv6 node after a reachability confirmation event occurs
ipv6 nd suppress-ra	Disables router advertisement transmissions

For additional information about using IPv6 profiles to configure dynamic interfaces, see *Creating an IPv6 Profile* in *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 2, Configuring IPv6* and *JUNOSe Link Layer Configuration Guide, Chapter 12, Configuring Dynamic Interfaces*.

RADIUS-Based Configuration

You can use RADIUS attribute Ipv6-NdRa-Prefix (VSA 26-129) to configure the prefix used in IPv6 Neighbor Discovery route advertisements. RADIUS then includes the VSA in Access-Accept messages. For information about the Ipv6-NdRa-Prefix RADIUS attribute, see *JUNOSe Broadband Access Configuration Guide, Chapter 3, Configuring RADIUS Attributes* and *JUNOSe Broadband Access Configuration Guide, Chapter 6, RADIUS Attribute Descriptions*.

ipv6 nd

- Use to enable the IPv6 Neighbor Discovery process on an interface.
- Example

  host1(config)#**interface fastEthernet 3/0**
  host1(config-if)#**ipv6 nd**

- Use the **no** version of this command to disable the Neighbor Discovery process.

ipv6 nd active-solicitations

- Use to specify that the router actively solicit neighbors that become stale (inactive). Normally, when a neighbor entry goes from a reachable state to a stale state, the router drops traffic until it resolves this neighbor entry. When enabled, the **ipv6 nd active-solicitations** command allows the router to use the stale neighbor entry while it solicits the neighbor. If the neighbor solicitation fails, the router removes the entry from the neighbor table and does not use the neighbor to forward any traffic.
- Example

  host1(config-if)#**ipv6 nd active-solicitations**

- Use the **no** version of this command to disable active solicitations.

ipv6 nd managed-config-flag

- Use to set the “managed address configuration” flag in IPv6 router advertisements.
- Example

  host1(config-if)#**ipv6 nd managed-config-flag**

- Use the **no** version of this command to clear the flag from IPv6 router advertisements.

ipv6 nd ns-interval

- Use to specify the interval, in milliseconds, between IPv6 neighbor solicitation retransmissions on an interface.
- Example

  host1(config-if)#**ipv6 nd ns-interval 500**

- Use the **no** version of this command to return the interval between neighbor solicitation retransmission to its default value (zero [0] milliseconds for router advertisements and 1000 milliseconds for Neighbor Discovery activity of the E-series router).

ipv6 nd other-config-flag

- Use to set the "other stateful configuration" flag in IPv6 router advertisements.
- Example

  host1(config-if)#**ipv6 nd other-config-flag**

- Use the **no** version of this command to clear the flag from IPv6 router advertisements.

ipv6 nd prefix-advertisement

- Use to specify which IPv6 prefixes the system includes in IPv6 router advertisements.
- Example

  host1(config-if)#**ipv6 nd prefix-advertisement 2002:1::/64 60000 45000 onlink autoconfig**

- Use the **no** version of this command to remove any prefixes from the IPv6 routing advertisements.

ipv6 nd ra-interval

- Use to specify the interval, in seconds, between IPv6 router advertisement retransmissions on an interface.
- Example

  host1(config-if)#**ipv6 nd ra-interval 500**

- Use the **no** version of this command to restore the default interval, 200 seconds.

ipv6 nd ra-lifetime

- Use to specify the router lifetime value, in seconds, in IPv6 router advertisements on an interface. The router lifetime value is the amount of time the router is considered the default router on this interface.
- Example

  host1(config-if)#**ipv6 nd ra-lifetime 900**

- Use the **no** version of this command to restore the default lifetime, 1800 seconds.

ipv6 nd reachable-time

- Use to specify the amount of time that the E-series router can reach a remote IPv6 node after some reachability confirmation event has occurred.
- Example 1—Sets the reachable-time to 30,000 milliseconds

  host1(config-if)#**ipv6 nd reachable-time 30000**

- Example 2—Sets the reachable-time to 1 hour, 10 minutes, and 45 seconds

  host1(config-if)#**ipv6 nd reachable-time 1 10 45**

- Use the **no** version of this command to restore the default value (zero [0] milliseconds for router advertisements and 3,600,000 milliseconds [1 hour] for Neighbor Discovery activity of the E-series router).

ipv6 nd suppress-ra

- Use to suppress IPv6 router advertisement transmissions on a local area network (Ethernet) interface.
- Example

  host1(config-if)#**ipv6 nd suppress-ra**

- Use the **no** version of this command to reenable the sending of IPv6 router advertisement transmissions on the LAN (Ethernet) interface

ipv6 nd suppress-ra-source-link-layer

- Use to suppress IPv6 router advertisement transmissions on a local area network (Ethernet) interface.
- Example

  host1(config-if)#**ipv6 nd suppress-ra-source-link-layer**

- Use the **no** version of this command to reenable the sending of IPv6 router advertisement transmissions on the LAN (Ethernet) interface.

Configuring Proxy Neighbor Advertisements

Much like proxy ARP, proxy Neighbor Discovery is a means by which one interface responds to a Neighbor Discovery query on behalf of another interface.

To configure proxy Neighbor Discovery:

1. Access an IPv6 interface.

   host1(config)#**interface fastEthernet 0/0**
   host1(config-if)#

2. Enable Neighbor Discovery on the current interface.

   host1(config)#**ipv6 nd**

NOTE: This command is redundant when configuring Neighbor Discovery over Ethernet, because neighbor solicitations and advertisements are automatically sent on Ethernet interfaces.

3. Enable IPv6 neighbor proxy.

   host1(config-if)#**ipv6 nd proxy**

ipv6 nd proxy

- Use to enable or disable Neighbor Discovery proxy.
- Example

  host1(config-if)#**ipv6 nd proxy**

- Use the **no** version of this command to disable Neighbor Discovery proxy.

Configuring Duplicate Address Detection Attempts

The duplicate address detection feature helps to verify that a new unicast IPv6 address is unique in the network. The router sends the IPv6 address in its neighbor solicitation messages. However, the router relies on the receiving device to understand the address duplication and does not prompt a conflict if the address already exists.

The CLI allows you to specify the number of consecutive neighbor solicitation messages that the router sends from the IPv6 interface.

ipv6 nd dad attempts

- Use to specify the number of consecutive neighbor solicitation messages that the router sends from the IPv6 interface.
- Use an attempt value of zero (0) to disable duplicate address detection on the current interface.
- The router suspends duplicate address detection on interfaces that are administratively down.
- Example

  host1(config-if)#**ipv6 nd dad attempts 10**

- Use the **no** version of this command to restore the default number of attempts to one (1).

Monitoring Neighbor Discovery

Neighbor Discovery-specific output appears in the output of various IPv6 **show** commands. For detailed information about IPv6 **show** commands and their output, see *Chapter 2, Configuring IPv6*.

Part 2

Internet Protocol Routing

Chapter 4
Configuring RIP

This chapter describes how to configure the Routing Information Protocol (RIP) on your E-series router; it contains the following sections:

Overview

RIP is an interior gateway protocol (IGP) typically used in small, homogeneous networks. RIP uses distance-vector routing to route information through IP networks.

Distance-vector routing requires that each router simply inform its neighbors of its routing table. For each network path, the receiving router picks the neighbor advertising the lowest metric, then adds this entry into its routing table for readvertisement.

Any host that uses RIP is assumed to have interfaces to one or more networks. These networks are considered to be directly connected networks. RIP relies on access to certain information about each of these networks. The most important information is the network's metric.

RIP Metric

RIP uses the hop count as the metric (also known as cost) to compare the value of different routes. The hop count is the number of routers that data packets must traverse between RIP networks. Metrics range from 0 for a directly connected network to 16 for an unreachable network. This small range prevents RIP from being useful for large networks.

RIP Messages

RIP exchanges routing information via User Datagram Protocol (UDP) data packets. Each RIP router sends and receives datagrams on UDP port number 520, the RIP version 1/RIP version 2 port. All communications intended for another router's RIP process area are sent from the RIP port.

Every RIP message contains a RIP header that consists of a command and a version number. The router supports RIP version 1 (RIPv1) and RIP version 2 (RIPv2) extensions.

RIP employs the following message types:

- Request—A request for the responding router to send all or part of its routing table.
- Response—A message containing all or part of the sender's routing table. This message is sent in response to a request or is an unsolicited routing update generated by the sender.

The RIP request and response messages also contain a list of route entries. Each route entry contains the following:

- Address Entry Identifier—The type of address
- Destination IP address—The destination address of the message
- Cost to reach the destination—A value between 1 and 15, which specifies the current metric for reaching the destination

Platform Considerations

For information about modules that support RIP on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support RIP.

For information about modules that support RIP on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support RIP.

References

For more information about RIP, consult the following resources:

- RFC 1058—Routing Information Protocol (June 1998)
- RFC 2453—RIP Version 2 (November 1998)

Features

Some of the major RIP features supported by the router include:

- authentication
- BFD liveness detection
- equal-cost multipath
- multicast addressing
- next hop
- poison reverse
- remote neighbors
- RIP version 1
- RIP version 2
- route summarization
- route tags
- split horizon
- subnet masks

Route Tags

A route tag is a field in a RIP message that allows boundary routers in an autonomous system (AS) to exchange information about external routes. Route tags provide a method of separating internal RIP routes (routes within the RIP routing domain) from external RIP routes, which may have been imported from an EGP (exterior gateway protocol) or another IGP (interior gateway protocol).

Routers supporting protocols other than RIP should be configurable to allow the route tags to be configured for routes imported from different sources. For example, routes imported from BGP should be able to have their route tags set to the number of the ASs from which the routes were learned.

Authentication

RIPv1 does not support authentication. If you are sending and receiving RIPv2 packets, you can enable RIP authentication on an interface.

The router provides the simple authentication scheme for RIPv2. Because authentication is a per message function and only one 2-octet field is available in the RIP message header, authentication uses the space of an entire RIP message.

The first 20-byte entry in a RIP authentication message contains an address family identifier value of 0xffff and a route tag value of 2. If the 0xffff address family is present in the RIP message, the remaining 16 octets of the entry contain a plain text password. If the password is fewer than 16 octets, it must be left-justified and padded to the right with nulls (0x00).

Authentication is applied per RIP interface. You can specify either **text** or **MD5** authentication. Text authentication uses a simple password that must be shared by the neighbors receiving updates or requests. If they do not have this password, the neighbors reject all updates or requests from the router. MD5 authentication uses a shared key to encrypt the RIP message. The neighbors must have the MD5 key to decrypt the message and encrypt a response.

NOTE: Do not use text authentication when security is important, because the router sends the unencrypted password in every RIP packet it sends.

Example 1 The following example shows how to use password authentication:

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#ip rip send version 2
host1(config-if)#ip rip authentication mode text
host1(config-if)#ip rip authentication key ke6G72mV
```

Example 2 The following example shows how to use MD5 authentication:

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#ip rip send version 2
host1(config-if)#ip rip authentication mode md5 8
host1(config-if)#ip rip authentication key sf43nBScE9
```

Subnet Masks

The Subnet Mask field of a RIP message contains the subnet mask that is applied to the IP address to set the nonhost portion of the address. If the subnet mask field in a RIP message contains a zero, then no subnet mask was included for the entry.

On an interface where a RIPv1 router may hear and operate on information in a RIPv2 routing entry, the following rules apply:

- Information internal to one network must never be advertised into another network.
- Information about a more specific subnet may not be advertised where RIPv1 routers would consider it a host route.
- Supernet routes (routes where a netmask is less specific than the natural network mask) must not be advertised where they could be misinterpreted by RIPv1 routers.

Next Hop

The Next Hop field in a RIP message contains the next IP address where a packet is sent. A value of zero in this field indicates that the next address the packet should be sent to is the router that originally sent the RIP message.

Multicasting

To reduce unnecessary load on hosts that are not listening to RIPv2 messages, an IP multicast address is used for periodic broadcast messages. The IP multicast address is 224.0.0.9.

Route Summaries

You can summarize routes reported by RIP to reduce the size of the routing table and the amount of traffic resulting from RIP updates. Configuring a RIP summary will cause that prefix to be advertised with the associated metric regardless of the presence of more-specific prefixes. Any more-specific prefixes will not be advertised when they are covered by the summary. You can choose the degree of summarization by using a prefix tree to specify the number of bits to report for routes matching a route map. Alternatively, you can explicitly specify routes for RIP to summarize.

Prefix Tree Example

The following example shows how to configure a 16-bit route summary:

1. Specify a route map for RIP in Router Configuration mode.

```
host1#configure t
Enter configuration commands, one per line. End with CNTL/Z.
host1(config)#router rip
host1(config-router)#route-map 1
host1(config-router)#exit
```

2. Define a route map associated with a prefix tree.

```
host1(config)#
host1(config)#route-map 1
host1(config-route-map)#match-set
host1(config-route-map)#match-set summary prefix-tree boston
host1(config-route-map)#exit
host1(config)#
```

3. Set the conditions for summarization in the prefix tree, including which routes are summarized and how many bits of the network addresses are preserved as the network prefix.

```
host1(config)#ip prefix-tree boston permit 2.1.0.0/16
```

This example summarizes routes for networks addressed by 2.1.*x*.*x*. The first 16 bits of the network address are preserved in the summary. For example, routes 2.1.3.0, 2.1.2.0, and 2.1.1.0 would all be summarized as 2.1.0.0.

Static Summary Example

You can use the **ip summary-address** command to specify routes that RIP will summarize.

```
host1(config-router)#ip summary-address 4.4.0.0 255.255.0.0 5
host1(config-router)#ip summary-address 4.3.0.0 255.255.0.0 6
```

Split Horizon

Split horizon is a mechanism to aid in preventing routing loops when distance-vector routing protocols such as RIP are employed in broadcast networks. When split horizon is enabled, the router cannot advertise information about routes on an interface from which the information originates. Split horizon is enabled by default on the router.

You can disable split horizon and enable poison reverse routing updates that advertise routes originating on the interface, but for each of these routes the metric is set to infinity to explicitly advertise that these networks are not reachable.

Equal-Cost Multipath

RIP supports equal-cost multipath (ECMP) and installs into the routing table multiple entries for paths to the same destination. Each of these multiple paths to a given destination must have the same cost as the others, but a different next hop.

Applying Route Maps

You can apply a policy to redistributed routes with the **route-map** command. See *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*, for more information about route maps. You can use the **table-map** command to apply a route map to RIP routes that are about to be added to the IP routing table.

Before You Run RIP

At least one IP address must be configured on your router for RIP to run.

Configuration Tasks

To configure RIP:

1. Create a RIP process by enabling RIP.

   host1(config)#**router rip**

2. (Optional) Configure the global RIP version. RIPv1 is used by default.

   host1(config-router)#**version 2**

3. (Optional) Do one of the following:

   - Associate a network with a RIP routing process and optionally configure RIP for the network.

     host1(config-router)#**network 10.2.1.0 255.255.255.0**
     host1(config-if)#**ip rip**
     host1(config-if)#**ip rip receive version 1**
     host1(config-if)#**ip rip send version 2**
     host1(config-if)#**ip rip authentication mode text**
     host1(config-if)#**ip rip authentication key klaatu42**

   - Associate the RIP routing process with an interface specified by an IP address or with an unnumbered interface, and configure RIP for the interface.

     host1(config-router)#**address 10.2.1.1**
     host1(config-router)#**address 10.2.1.1 receive version 1**
     host1(config-router)#**address 10.2.1.1 send version 2**
     host1(config-router)#**address 10.2.1.1 authentication mode text**
     host1(config-router)#**address 10.2.1.1 authentication key 31barada**

   Each configuration step is optional, and includes the following:

   - (Optional) Specify a RIP receive version for an interface. By default, RIP interfaces on your router receive both RIPv1 and RIPv2.

   - (Optional) Specify a RIP send version for an interface. By default, RIP interfaces on your router send only RIPv1.

   - (Optional) Specify an authentication mode and authentication password or key. This step is permitted only if both receive version and send version are set to RIPv2.

4. (Optional) Enable RIP to advertise a default route.

   host1(config-router)#**default-information originate**

5. (Optional) Specify a default metric for redistributed routes on all subsequently created interfaces.

   host1(config-router)#**default-metric 5**

6. (Optional) Set the administrative distance for advertised routes.

   host1(config-router)#**distance 150**

7. (Optional) Control the dynamic distribution of routes caused by changes to an associated route map.

   host1(config-router)#**disable-dynamic-redistribute**

8. (Optional) Adjust RIP timers.

   host1(config-router)#**timers update 20**
   host1(config-router)#**timers invalid 60**
   host1(config-router)#**timers holddown 60**
   host1(config-router)#**timers flush 90**

9. (Optional) Specify maximum number of ECMP paths.

   host1(config-router)#**maximum-paths 2**

10. (Optional) Summarize routes.

    Use a prefix tree to specify the number of bits to report for routes matching a route map:

    host1(config)#**ip prefix-tree boston permit 10.10.2.0/24**
    host1(config-router)#**route-map 4**
    host1(config-route-map)#match-set summary prefix-tree boston

NOTE: For information about the **ip prefix-tree** command, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

    Alternatively, explicitly specify routes for RIP to summarize:

    host1(config-router)#**ip summary-address 4.4.0.0 255.255.0.0 5**
    host1(config-router)#**ip summary-address 4.3.0.0 255.255.0.0 6**

11. (Optional) Redistribute routes from other protocols into RIP, or from RIP to other protocols.

    host1(config-router)#**redistribute rip 5**
    host1(config-router)#**route-map 4**
    host1(config-router)#**redistribute bgp 100 route-map 4**

12. (Optional) Enable unicast communication with RIP neighbors.

    host1(config-router)#**neighbor 10.10.21.100**
    host1(config-router)#**passive-interface atm atm 2/0.16**

13. (Optional) Set the debounce time for interfaces brought down by some event.

    host1(config-router)#**debounce-time 30**

14. (Optional) Prevent RIP from purging the routing table for interfaces brought down by some event.

    host1(config-router)#**interface-event-disable**

15. (Optional) Prevent RIP from sending a more-specific route if a less-specific route has a better metric.

    host1(config-router)#**send-more-specific-routes-disable**

16. (Optional) Prevent RIP from sending triggered updates.

    host1(config-router)#**triggered-update-disable**

17. (Optional) Apply a table map to modify route distance.

    host1(config-router)#**table-map dist1**

Relationship Between address and network Commands

If you use the **network** command to configure a RIP network, use the **ip rip** commands to configure the RIP attributes for that network. Do not use the **address** commands.

If you use the **address** command to configure a RIP network, use the **address** commands to configure the RIP attributes for that network. Do not use the **ip rip** commands.

NOTE: The **network** and **ip rip** commands are maintained for industry compatibility. You can configure all your RIP interfaces with the **address** commands. You cannot configure unnumbered interfaces with the **network** and **ip rip** commands.

address

- Use to configure RIP to run on the interface specified by the IP address or on an unnumbered interface. Use the **address** commands to configure RIP attributes on the network.
- Configures RIP with the default values: Send version is RIPv1, receive version is RIPv1 and RIPv2, authentication is not enabled.
- Example

  host1(config-router)#**address 10.2.1.1**

- Use the **no** version to delete the RIP interface.

address authentication key

- Use to specify either the simple password for text authentication or the encryption/decryption key for MD5 authentication. The key is a string of up to 16 alphanumeric characters and can be mixed uppercase and lowercase.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**address 10.2.1.1 authentication key ke6G72mV**

- Use the **no** version to clear all authentication keys.

address authentication mode

- Use to specify the authentication mode.
- Specify **text** to send a simple text password to neighbors. If a neighbor does not have the same password, requests and updates from this router are rejected.
- Specify **md5** *keyID* to send an MD5 hash to neighbors. Neighbors must share the MD5 key to decrypt the message and encrypt the response.
- Example

  host1(config-router)#**address 10.2.1.1 authentication mode text**

- Use the **no** version to remove authentication from all RIP interfaces.

address receive version

- Use to restrict the RIP version that the router can receive on an interface. The default is to receive both RIPv1 and RIPv2.
- Example

  host1(config-router)#**address 10.2.1.1 receive version 1**

- Use the **no** version to restore the default value, 1 2.

address send version

- Use to restrict the RIP version that the router can send on an interface. The default is to send only RIPv1.
- Example

  host1(config-router)#**address 10.2.1.1 send version 2**

- Use the **no** version to restore the default value, 1.

clear ip rip redistribution

- Use to clear all the routes that have previously been redistributed into RIP.
- Example

  host1#**clear ip rip redistribution**

- There is no **no** version.

debounce-time

- Use to control the interval RIP waits before bringing back up an interface that was brought down by some event.
- The interval can be in the range 0–60 seconds.
- Example

  host1(config-router)#**debounce-time 30**

- Use the **no** version to restore the default value, 10 seconds.

default-information originate

- Use to enable RIP to advertise a default route (0.0.0.0/0) if the default route exists in the IP routing table.
- If the default route does not exist, you must configure it using the **ip route** command, or specify the **always** keyword. The **always** keyword causes RIP to always advertise the default route, and creates it if it is not present in the IP routing table.
- Example

  host1(config-router)#**default-information originate**

- Use the **no** version to disable advertisement of the default route.

default-metric

- Use to configure RIP to apply this metric for redistributed routes on all subsequently created interfaces.
- Configuring a default metric lowers the priority of the routes.
- Use a metric in the range 1 – 16.
- Example

  host1(config-router)#**default-metric 5**

- Use the **no** version to restore the default value, 0.

disable

- Use to disable RIP processing.
- Example

  host1(config-router)#**disable**

- Use the **no** version to enable RIP processing.

disable-dynamic-redistribute

- Use to halt the dynamic redistribution of routes that are initiated by changes to a route map.
- Dynamic redistribution is enabled by default.
- Example

  host1(config-router)#**disable-dynamic-redistribute**

- Use the **no** version to reenable dynamic redistribution.

distance

- Use to set the administrative distances for routes.
- Example

  host1(config-router)#**distance 150**

- Use the **no** version to restore the default value, 120.

distribute-list

- Use to apply a specific access list to incoming or outgoing RIP route updates.
- An IP access list acts as a filter. Refer to the **access list** command in the *JUNOSe Command Reference Guide A to M* for more information.
- Example

  host1(config-router)#**distribute-list 5 incoming**

- Use the **no** version to stop application of the distribute list.

interface-event-disable

- Use to configure RIP to purge the routing table for interfaces that were brought down by some event.
- Example

  host1(config-router)#**interface-event-disable**

- Use the **no** version to restore the default condition, wherein RIP does not automatically purge the routing table for down interfaces.

ip rip

- Use to configure RIP on the network interface specified with the **network** command.
- Configures RIP with the default values: Send version is RIPv1, receive version is RIPv1 and RIPv2, authentication is not enabled.
- Example

  host1(config-if)#**ip rip**

- Use the **no** version to delete the RIP interface.

ip rip authentication key

- Use to specify either the simple password for text authentication or the encryption/decryption key for MD5 authentication. The key is a string of up to 16 alphanumeric characters and can be mixed uppercase and lowercase.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-if)#**ip rip authentication key ke6G72mV**

- Use the **no** version to clear all authentication keys.

ip rip authentication mode

- Use to specify the authentication mode.
- Specify **text** to send a simple text password to neighbors. If a neighbor does not have the same password, requests and updates from this router are rejected.
- Specify **md5** *keyID* to send an MD5 hash to neighbors. Neighbors must share the MD5 key to decrypt the message and encrypt the response.
- Example

  host1(config-if)#**ip rip authentication mode text**

- Use the **no** version to remove authentication from all RIP interfaces.

ip rip receive version

- Use to restrict the RIP version that the router can receive on an interface. The default is both RIPv1 and RIPv2.
- Example

  host1(config-if)#**ip rip receive version 1**

- Use the **no** version to restore the default value, 1 2.

ip rip send version

- Use to restrict the RIP version that the router can send on an interface. The default is RIPv1.
- Example

  host1(config-if)#**ip rip send version 2**

- Use the **no** version to restore the default value, 1.

ip split-horizon

- Use to configure the split horizon feature and poison reverse features for the interface. Enabled by default, split horizon prevents the RIP router from advertising routes from the originating interface.
- Poison reverse routing updates are disabled by default; when enabled, they set the metric for routes originating on the interface to infinity, thus explicitly advertising that the network is not reachable. This helps to prevent routing loops.
- In most configurations, you will want to accept the default condition.
- Example

  host1(config-if)#**no ip split-horizon**

- Use the **no** version to disable split horizon and enable poison reverse routing updates.

ip summary-address

- Use to specify an IP address and network mask to identify which routes to summarize.
- You can optionally specify a metric associated with the summary address. The default metric is 1.
- Example

  host1(config-router)#**ip summary-address 4.4.0.0 255.255.0.0 5**
  host1(config-router)#**ip summary-address 4.3.0.0 255.255.0.0 6**

- Use the **no** version to stop summarization for the specified routes.

match-set summary prefix-tree

- Use to specify a prefix tree that summarizes routes for a particular route map.
- Use the **ip prefix-tree** command to set the conditions of the prefix tree, including which routes to summarize and how many bits of the network address to preserve.
- Example

  host1(config-route-map)#**match-set summary prefix-tree boston**

- Use the **no** version to disable the use of the prefix tree by the route map.

maximum-paths

- Use to control the maximum number of parallel routes that RIP can support.
- RIP installs multiple equal-cost paths to a given destination only if each has a different next hop.
- The maximum number of routes can be in the range 1–16.
- Example

  host1(config-router)#**maximum-paths 2**

- Use the **no** version to restore the default value, 4.

neighbor

- Use to specify a RIP neighbor to which the router sends unicast messages.
- You must also use the **passive-interface** command to specify the interface as passive, thereby restricting the interface to unicast RIP messages.
- Example

  host1(config-router)#**neighbor 10.10.21.100**

- Use the **no** version to remove the neighbor.

network

- Use to associate a network with a RIP routing process. Use the **ip rip** commands to configure RIP attributes on the network.
- You supply a network mask to the new address so that RIP runs on that specific network.
- If you do not specify an interface's network, the network is not advertised in any RIP updates.
- You can specify either the standard subnet mask or the inverse subnet mask.
- Example 1—standard subnet mask

  host1(config-router)#**network 10.2.1.0 255.255.255.0**

- Example 2—inverse subnet mask

  host1(config-router)#**network 10.2.1.0 0.0.0.255**

- Use the **no** version to disable RIP on the specified interface.

passive-interface

- Use to disable the transmission of multicast RIP messages on the interface.
- RIP messages are unicast to a RIP neighbor on the interface if the interface is present in the IP routing table as the next-hop interface to the configured neighbor.
- Example

  host1(config-router)#**passive-interface atm atm 2/0.16**

- Use the **no** version to reenable the transmission of RIP multicast messages on the specified interface.

redistribute

- Use to redistribute information from a routing domain other than RIP into the RIP domain.
- Specify the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **isis**, **ospf**, **static** [**ip**], and **connected**. Use the **static** keyword to redistribute IP static routes; optionally add the **ip** keyword when redistributing into IS-IS. The keyword **connected** refers to routes that are established automatically by virtue of having enabled IP on an interface. For routing protocols such as OSPF and IS-IS, these routes will be redistributed as external to the AS.
- Use the **route-map** keyword to interrogate the route map to filter the importation of routes from the source routing protocol to the current routing protocol. If you do not specify the route-map option, all routes are redistributed. If you specify the route-map option, but no route map tags are listed, no routes will be imported.
- Use to redistribute routes from RIP into other non-RIP routing domains.
- Example 1

  host1(config)#**router rip 5**
  host1(config-router)#**redistribute bgp 100 route-map 4**

- Example 2

  host1(config)#**router bgp 100**
  host1(config-router)#**redistribute rip 5**

- Use the **no** version to disable redistribution.

route-map

- Use to specify a route map for RIP.
- Example

  host1(config)#**router rip**
  host1(config-router)#**route-map 4**

- Use the **no** version to delete the route map. If you do not specify an interface, it removes the global route map if it exists.

router rip

- Use to enable RIP routing protocol and specify a RIP process for IP, or to access Router Configuration mode.
- Specify only one RIP process per router.
- Example

  host1(config)#**router rip**

- Use the **no** version to delete the RIP process and removes the configuration from your router.

send-more-specific-routes-disable

- Use to configure RIP to send a less-specific route in preference to a more-specific route if the less-specific route has a metric.
- Example

  host1(config-router)#**send-more-specific-routes-disable**

- Use the **no** version to restore the default condition, wherein RIP always sends a more-specific route in preference to a less-specific route, even if the less-specific route has a metric.

table-map

- Use to apply a policy to modify distance, metric, or tag values of RIP routes about to be added to the IP routing table.
- The new route map is applied to all routes currently in and those subsequently placed in the forwarding table. Previously redistributed routes are redistributed with the changes caused by the route map.
- To remove from the forwarding table any old routes that are now disallowed by the specified route map, you must refresh the IP routing table with the **clear ip routes *** command.
- Example

  host1(config)#**route-map dist1 permit 5**
  host1(config-route-map)#**match community boston42**
  host1(config-route-map)#**set distance 33**
  host1(config-route-map)#**exit**
  host1(config)#**router rip 100**
  host1(config-router)#**table-map dist1**
  host1(config-router)#**exit**
  host1(config)#**exit**
  host1#**clear ip routes ***

- Use the **no** version to halt application of the route map.

timers

- Use to configure RIP timers.
- The router supports the following RIP timers:
  - update—Interval in seconds at which routing updates are sent. The default is 30 seconds.
  - invalid—Interval in seconds after which a route is declared invalid (null). Set this value to at least three times the update value. The default is 180 seconds.
  - holddown—Interval in seconds during which routing information about better paths is suppressed. Set this value to at least three times the update value. The default is 120 seconds.
  - flush—Interval in seconds that must pass before a route is removed from the routing table. Set this value greater than the invalid value. The default is 300 seconds.
- Example

  host1(config-router)#**timers update 20**
  host1(config-router)#**timers invalid 60**
  host1(config-router)#**timers holddown 60**
  host1(config-router)#**timers flush 90**

- Use the **no** version to restore the default values, 30 180 120 300.

triggered-update-disable

- Use to prevent RIP from sending triggered routing updates.
- Example

  host1(config-router)#**triggered-update-disable**

- Use the **no** version to restore the default condition, wherein RIP does sends triggered routing updates.

version

- Use to specify the global RIP version. The default is RIPv1.
- To change the RIP version on a specific interface, use the **ip rip receive version** and the **ip rip send version** commands, or the **address receive version** and **address send version** commands.
- Example

  host1(config-router)#**version 2**

- Use the **no** version to revert to the default value, 1.

Enabling RIP on Dynamic IP Interfaces

You can use the **ip rip copy-to-dynamic** command to enable RIP on dynamic, unnumbered IP interfaces. This command allows the dynamic interfaces, as they are created, to copy RIP settings from a numbered IP interface to which the interfaces refer for their source address.

CAUTION: RIP transmits a complete set of routing updates at each update interval. This can result in a very large number of RIP updates. When configuring RIP over dynamic interfaces, we strongly recommend that you configure an output policy on the reference interface to limit the amount of routing information that RIP transmits to each peer.

ip rip copy-to-dynamic

- Use to enable RIP on dynamic, unnumbered IP interfaces. This command allows the dynamic interfaces to copy RIP settings from the numbered IP interface to which the interfaces refer for their source address.
- Once created, the dynamic RIP interfaces do not track configuration changes on the numbered interface from which they originally inherited the configuration. To reinherit RIP settings, use the **clear ip rip dynamic-interfaces** command.

CAUTION: Issuing the **ip rip copy-to-dynamic** command enables RIP on all dynamic unnumbered interfaces that reference the interface and become active after issuing the command. This may unintentionally include dynamic interfaces created on MPLS tunnels or subscriber interfaces where you would not want to enable RIP. To avoid this possible misconfiguration, take care to reference dynamic interfaces where RIP is not required to another numbered interface on which RIP is not enabled.

- Example

  host1(config-if)#**ip rip copy-to-dynamic**

- Use the **no** version to stop the use of RIP configuration on any new, dynamic, unnumbered IP interfaces. The **no** version does not remove all existing, active RIP interfaces that were created after issuing this command. To remove all existing, active RIP interfaces, use the **no rip copy-to-dynamic** command to stop the use of RIP on any new, dynamic interfaces, and then use the **clear ip rip dynamic-interfaces** command to clear any existing RIP dynamic interfaces.

Clearing Dynamic RIP Interfaces

You can use the **clear ip rip dynamic-interfaces** to clear any existing dynamic RIP interfaces that were created by the **ip rip copy-to-dynamic** command. If the router is still using the **ip rip copy-to-dynamic** command, when the router recreates the dynamic interfaces, they use the RIP attributes from the interface to which they refer. If the router no longer uses the **ip rip copy-to-dynamic** command, any newly created dynamic interfaces do not use the RIP attributes from the reference interface.

clear ip rip dynamic-interfaces

- Use to clear all existing dynamic, unnumbered interfaces that were created since issuing the **ip rip copy-to-dynamic** command.
- Example

  host1#**clear ip rip dynamic-interfaces**

- There is no **no** version.

Using RIP Routes for Multicast RPF Checks

You can use the **ip route-type** command to specify whether RIP routes are available for only unicast forwarding protocols or only multicast reverse path forwarding (RPF) checks. Routes available for unicast forwarding appear in the unicast view of the routing table, whereas routes available for multicast RPF checks appear in the multicast view of the routing table.

ip route-type

- Use to specify whether RIP routes are available only for unicast forwarding, only for multicast reverse path forwarding checks, or for both.
- Use the **show ip route** command to view the routes available for unicast forwarding.
- Use the **show ip rpf-routes** command to view the routes available for multicast reverse path forwarding checks.
- By default, RIP routes are available for both unicast forwarding and multicast reverse path forwarding checks.
- Example

  host1(config)#**router rip**
  host1(config-router)#**ip route-type unicast**

- Use the **no** version to restore the default value, both.

Configuring the BFD Protocol for RIP

The **address bfd-liveness-detection** command or the **ip rip bfd-liveness-detection** command configures the Bidirectional Forwarding Detection (BFD) protocol for RIP. The BFD protocol uses control packets and shorter detection time limits to more rapidly detect failures in a network. Also, because they are adjustable, you can modify the BFD timers for more or less aggressive failure detection.

Without BFD, when a RIP peer goes down, the routes learned from that peer are purged only after each route times out. The timeout is configurable with the **timers invalid** command. By default, the timeout is 180 seconds after each route was received or refreshed. Consequently routes are purged successively over varying time periods rather than all at once.

In contrast, when a BFD session exists between RIP peers, a peer that goes down is detected quickly. RIP simultaneously purges all routes learned from that peer and starts the hold-down timer for each peer.

When you issue the **address bfd-liveness-detection** command or the **ip rip bfd-liveness-detection** command on a RIP peer, the peer establishes BFD liveness detection with all BFD-enabled RIP peers. When the local peer receives an update from a remote RIP peer—if BFD is enabled and if the session is not already present—the local peer attempts to create a BFD session to the remote peer.

Each adjacent pair of peers negotiates an acceptable transmit interval for BFD packets. The negotiated value can be different on each peer. Each peer then calculates a BFD liveness detection interval. When a peer does not receive a BFD packet within the detection interval, it declares the BFD session to be down and purges all routes learned from the remote peer.

NOTE: Before the router can use the **address bfd-liveness-detection** command or the **ip rip bfd-liveness-detection** command, you must specify a BFD license key. To view an already configured license, use the **show license bfd** command.

For general information about configuring and monitoring the BFD protocol, see *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.

address bfd-liveness-detection
ip rip bfd-liveness-detection

- Use to enable BFD (bidirectional forwarding detection) and define BFD values to more quickly detect RIP data path failures.
- Use the **address bfd-liveness-detection** command when you have used the address command to configure the RIP network. Use the **ip rip bfd-liveness-detection** command when you have used the **network** command to configure the RIP network.
- The peers in a RIP adjacency use the configured values to negotiate the actual transmit intervals for BFD packets.
  - You can use the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-interval** keyword to specify the same value for both of those intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value. The default value is 300 milliseconds.
- You can use the **multiplier** keyword to specify the detection multiplier value. The calculated BFD liveness detection interval can be different on each peer. The multiplier value is roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. The default value is 3.
- For details on liveness detection negotiation, see *Negotiation of the BFD Liveness Detection Interval* section in *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.
- You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.
- Example

  host1(config-if)#**ip rip bfd-liveness-detection minimum-interval 800**
  or
  host1(config-router)#**address bfd-liveness-detection minimum-interval 800**

- Use the **no** version to disable BFD on the RIP interface.

Remote Neighbors

You can create RIP remote neighbors to enable the router to establish neighbor adjacencies through unidirectional interfaces, such as MPLS tunnels, rather than the standard practice of using the same interface for receipt and transmission of RIP packets. The remote neighbor can be more than one hop away through intermediate routes that are not running RIP. RIP uses the interface associated with the best route to the remote neighbor to reach the neighbor. A best route to the neighbor must exist in the IP routing table.

You must explicitly configure remote neighbors on the RIP routers to specify the remote neighbor with which the router will form an adjacency and the source IP address the router will use for RIP packets destined to its peer remote neighbor.

To form an adjacency with its remote neighbor, the router sends all RIP packets to the remote neighbor as unicast packets with the destination IP address equal to the source IP address of the remote neighbor. The loopback interface associated with the source IP address for the remote neighbor acts as a logical RIP interface for the neighbor.

To prevent routing loops, you can disable split horizon and enable poison reverse routing updates.

The **remote-neighbor** command to specify the remote neighbors is mandatory. Configuration of all other remote-neighbor attributes is optional.

authentication key

- Use to specify the password for text authentication and the key for MD5 authentication for RIP remote-neighbor interface.
- This command is supported only in RIPv2. Authentication is disabled by default.
- Example

  host1(config-router-rn)#**authentication key 0 jun27ior**

- Use the **no** version to clear the key for the remote-neighbor interface.

authentication mode

- Use to specify the authentication mode for the remote neighbor interface.
- Specify **text** to send a simple text password to remote neighbors. If a remote neighbor does not have the same password, requests and updates from this router are rejected.
- Specify **md5** *keyID* to send an MD5 hash to remote neighbors. Remote neighbors must share the MD5 key to decrypt the message and encrypt the response.

- This command is supported only in RIPv2. Authentication is disabled by default.
- Example

  host1(config-router-rn)#**authentication mode text**

- Use the **no** version to remove authentication from the RIP remote-neighbor interface.

distribute-list

- Use to apply a specific access list to either incoming or outgoing RIP route updates on the RIP remote-neighbor interface.
- An IP access list acts as a filter. Refer to the **access list** command in the *JUNOSe Command Reference Guide A to M* for more information.
- Example

  host1(config)#**distribute-list 5 in**

- Use the **no** version to stop application of the distribute list.

exit-remote-neighbor

- Use to exit from the Remote Neighbor Configuration mode and return to Router Configuration mode.
- Example

  host1(config-router-rn)#**exit-remote-neighbor**

- There is no **no** version.

receive version

- Use to restrict the RIP version that the router can receive on a RIP remote-neighbor interface. The default is to receive both RIPv1 and RIPv2.
- The **off** keyword overrides any other specified option; for example, configuring both **1** and **off** or both **2** and **off** has the same result as configuring only **off**.
- Example

  host1(config-router-rn)#**receive version 1**

- Use the **no** version to restore the default value, 1 2.

remote-neighbor

- Use to configure a RIP remote neighbor.
- Example

  host1(config-router)#**remote-neighbor 10.25.100.14**

- Use the **no** version to remove the remote neighbor and any attributes configured for the remote neighbor.

send version

- Use to restrict the RIP version that the router can send on an interface. The default is to send only RIPv1.
- Example

  host1(config-router-rn)#**send version 1**

- Use the **no** version to restore the default value, 1.

split-horizon

- Use to configure the split horizon and poison reverse features for RIP remote neighbors.
- Split horizon is enabled by default; poison reverse routing updates are disabled by default.
- Poison reverse routing updates set the metric for routes originating on the interface to infinity, thus explicitly advertising that the network is not reachable. This helps to prevent routing loops.
- Example

  host1(config-router-rn)#**no split-horizon**

- Use the **no** version to disable the split horizon and enable poison reverse routing updates.

time-to-live

- Use to configure a hop count by setting the value of the time-to-live field used by packets sent to a RIP remote neighbor.
- Example

  host1(config-router-rn)#**time-to-live 12**

- Use the **no** version to restore the default value, 16.

update-source

- Use to specify the RIP interface whose local address is used as the source address for the RIP connection to a remote neighbor.
- The source address assigned to a remote neighbor must be unique. If you configure a RIP router to form neighbor adjacencies with two RIP remote neighbors, then the RIP router must have two unique local source IP addresses, one for each of its remote neighbors.
- Example

  host1(config-router-rn)#**update-source atm 2/0.17**

- Use the **no** version to delete the source address from the connection to the remote neighbor.

Monitoring RIP

Two sets of commands enable you to monitor RIP operation on your router: the **debug** and the **show** commands. Both sets of commands provide information about your router's RIP state and configuration.

The task you are performing with each of these monitoring commands is basically the same for each command; that is, you are requesting information. The results of this request may vary. For instance, the **debug** commands provide information about problems with the network or the router, whereas the **show** commands provide information about the actual state and configuration of your router.

debug Commands

The **debug** commands provide information about the following RIP items:

- General events, such as creating a RIP process or removing RIP from an interface
- Routing events, such as when two RIP routers exchange routes

debug ip rip

- Use to display information about selected RIP events. This command has many keywords that allow you to specify a variety of RIP events.
- You can set the level of severity for the events you want displayed; specify the desired descriptive term or a corresponding number (0–7).
- You can set the verbosity of the messages you want displayed: low, medium, high.
- Example

  host1#**debug ip rip events**

- Use the **no** version to cancel the display of any information about the designated variable.

undebug ip rip

- Use to cancel the display of information about a selected event.
- The same RIP variables can be designated as in the **debug ip rip** command.
- Example

  host1#**undebug ip rip events**

- There is no **no** version.

show Commands

Use the **show** commands to monitor the following types of RIP information:

- Configuration
- IP-related information
- Global counters
- Counters for a specified network
- Statistics

You can set a statistics baseline for RIP interfaces by using the **baseline ip rip** command.

You can specify a VRF instance for the **show ip rip** commands. You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

baseline ip rip

- Use to set a statistics baseline for RIP interfaces.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the optional **delta** keyword with the **show ip rip statistics** command to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ip rip**

- There is no **no** version.

show ip rip

- Use to display RIP information.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Use the **ifconfig** keyword to display address and interface configuration information instead of the default operational data.
- Field descriptions
  - Router Information Protocol Fields
    - Router Administrative State—Displays the RIP state. Enable means the router is allowed to send and receive updates. Disable means that RIP might be configured but it is not allowed to run yet.
    - System version RIP1—RIP versions allowed for sending and receiving RIP updates. The router version is currently set to RIP1, which sends RIPv1 but will receive RIPv1 or RIPv2. If it is set to RIP2, it will send and receive RIPv2 only. The default is configured for RIP1.
    - Incoming filters—Access list applied to incoming route updates

- Outgoing filters—Access list applied to outgoing route updates
- Global route map—Route map that specifies all RIP interfaces on the router
- Default metric—Value for redistributed routes. The default is 1. This global value is superseded by metrics applied to a RIP interface.
- Distance—Value added to RIP routes added to the IP routing table. The default is 120.
- Number of route changes—Number of times the router has been told to route changes by its peers
- Number of route queries—Number of times the router has received route requests from other routers
- Update interval—Current setting of the update timer (in seconds)
- Invalid interval—Current setting of the invalid timer (in seconds)
- Hold down time—Current setting of the hold-down timer (in seconds)
- Flush interval—Current setting of the flush timer (in seconds)
- Route Type—Whether RIP routes are available only for unicast forwarding, only for multicast reverse path forwarding checks, or for both
- Max Ecmp Paths—Number of parallel routes that RIP can support
- Default-Information originate always—Ability (enabled or disabled) of RIP to advertise a default route (0.0.0.0/0) if the default route exists in the IP routing table
- Triggered Updates—Ability (enabled or disabled) of RIP to send triggered updates
- Purge Routes on Interface Down Event—Ability (enabled or disabled) of RIP to purge the routing table for interfaces that were brought down by some event
- Send More Specific Routes—Ability (enabled or disabled) of RIP to send a less-specific route in preference to a more-specific route if the less-specific route has a metric
- Debounce Time—Debounce time for interfaces brought down by some event
- Default-Information originate—Ability (enabled or disabled) of RIP to advertise a default route (0.0.0.0/0) if the default route exists in the IP routing table
- route-map—Name of the route map specified for RIP
- Summary Address—Route that RIP summarizes
- Network—IP address of a network on which RIP is running
- Netmask—Network mask applied to the network address
- Neighbor—Configured neighbor information

- Address Operational Data
  - Unnumbered status—Status of the unnumbered interface
  - Received bad packet—Number of bad packets received
  - Received bad routes—Number of bad routes received
  - Triggered updates sent—Number of triggered updates sent; triggered updates are sent before the entire RIP routing table is sent; triggered by events such as adding a new RIP route or redistribution
  - Received updates—Number of updates received
  - Numbered status—Status of the numbered interface from which this interface obtains its configuration
  - Send version—Version of RIP used for sending updates
  - Receive version—Version of RIP accepted in received updates
  - Authentication mode—Password or MD5 authentication, or none
  - Default metric—Metric value applied to the RIP interface. The default is 1.
  - BFD minimum receive interval(msec)—Configured minimum interval requested between BFD control packets sent by the remote RIP peer; used with RIP peers to negotiate a detection interval for BFD session failure. The default is 300 milliseconds.
  - BFD minimum transmit interval(msec)—Configured minimum interval between BFD control packets sent by the local RIP peer; used with RIP peers to negotiate a detection interval for BFD session failure. The default is 300 milliseconds.
  - BFD multiplier—Multiplied by the negotiated BFD minimum receive interval to determine the interval between packets permitted before the BFD session is declared down. Also, the number of BFD control packets that the RIP local peer can miss before the BFD session is declared down. The default is 3.
  - Passive Interface—Whether or not the interface is passive, thereby restricting the interface to unicast RIP messages
  - Passive Interface—Whether or not the interface is passive, thereby restricting the interface to unicast RIP messages
  - Access-list applied to outgoing route—Name of the access list applied to outgoing routes
  - Access-list applied to incoming route—Name of the access list applied to incoming routes
  - Route-map applied to outgoing route—Name of the route map applied to outgoing routes

- Example 1

```
host1#show ip rip
Routing Information Protocol
 Router Administrative State = enable
 System version RIP2: send = 2, receive = 2
 No filter is applied to outgoing route update for all interfaces
 No filter is applied to incoming route update for all interfaces
 No global route map
```

```
 No table map
 Default metric = 1
 Distance = 120
 Number of route changes = 3
 Number of route queries = 0
 Update interval = 30 (secs)
 Invalid interval = 180 (secs)
 Hold down time = 120 (secs)
 Flush interval = 300 (secs)
 Route Type     =  both unicast and multicast
 Max Ecmp Paths = 4
 Default-Information originate always = enabled
 Triggered Updates = enabled
 Purge Routes on Interface Down Event  = enabled
 Send More Specific Routes  = enabled
 Debounce Time = 10
 Default-Information originate : disabled
     route-map :  none
  Summary Address:  None
 Network         netmask
  Neighbor
    No Configured Neighbors

          *** Address Operational Data ***

Unnumbered, Rip is up, ATM2/1.18
  Dynamic creation and inherits configuration from loopback1
  Received bad packet = 0
  Received bad routes = 0
  Triggered updates sent = 0
  Received updates = 9
1.1.1.1, Rip is up, loopback1
  Send version = 2
  Receive version = 2
  Authentication mode = none
  Default metric = 1
  Passive Interface = No
  Access-list applied to outgoing route = none
  Access-list applied to incoming route = none
  Route-map applied to outgoing route = none
  Copy configuration to dynamic interfaces
  Received bad packet = 0
  Received bad routes = 0
  Triggered updates sent = 0
  Received updates = 0
```

- Example 2

```
host1#show ip rip ifconfig
Routing Information Protocol
 Router Administrative State = enable
 System version RIP2: send = 2, receive = 2
 No filter is applied to outgoing route update for all interfaces
 No filter is applied to incoming route update for all interfaces
 No global route map
 No table map
 Default metric = 1
 Distance = 120
 Number of route changes = 17
 Number of route queries = 2
 Update interval = 30 (secs)
 Invalid interval = 180 (secs)
 Hold down time = 120 (secs)
```

```
 Flush interval = 300 (secs)
 Route Type     =  both unicast and multicast
 Max Ecmp Paths = 4
 Default-Information originate always = enabled
 Triggered Updates = enabled
 Purge Routes on Interface Down Event  = enabled
 Send More Specific Routes  = enabled
 Debounce Time = 10
 Default-Information originate : disabled
     route-map :  none
  Summary Address:  None
 Network         netmask
  Neighbor
    No Configured Neighbors

          *** Interface Configuration Data***

loopback1
  Send version = def
  Receive version = def
  Authentication mode = none
  Default metric = default
  Passive Interface = No
  Access-list applied to outgoing route = none
  Access-list applied to incoming route = none
  Route-map applied to outgoing route = none
  Copy configuration to dynamic interfaces

          *** Address Configuration Data ***

Unnumbered, Rip is up, ATM2/1.18
  Dynamic creation and inherits configuration from loopback1
  Received bad packet = 0
  Received bad routes = 0
  Triggered updates sent = 0
  Received updates = 3
1.1.1.1, Rip is up, loopback1
  Send version = def
  Receive version = def
  Authentication mode = none
  Default metric = default
  Passive Interface = No
  Access-list applied to outgoing route = none
  Access-list applied to incoming route = none
  Route-map applied to outgoing route = none
  Received bad packet = 0
  Received bad routes = 0
  Triggered updates sent = 0
  Received updates = 0
```

- Example 3—Interface configuration data excerpt showing BFD information.

```
host1#show ip rip ifconfig
...
   *** Interface Configuration Data***

FastEthernet1/0
  Send version = def
  Receive version = def
  Authentication mode = none
  Default metric = default
  BFD minimum receive interval(msec) =  400
  BFD minimum transmit interval(msec)= 500
```

```
BFD multiplier = 2
Passive Interface = No
Access-list applied to outgoing route = none
Access-list applied to incoming route = none
Route-map applied to outgoing route = none
```

show ip rip brief

- Use to display limited RIP information.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Field descriptions
  - IP Address—IP address of the interface where RIP is running
  - Tx—Transmit version of RIP on this interface, which can override the router configuration
  - Rx—Receive version of RIP on this interface
  - Auth—Type of authentication, password (text) or MD5
  - Met—Current value is the same as the router one (the default metric). Based on MIB 2 for RIP, the interface's route metric can be set individually.
  - AccList O/I—Access list applied to outgoing/incoming RIP route updates
  - RtMap—Identifier for the route map that specifies a summary of RIP routes
  - Status—Status of RIP, either up or down
  - Intf—Interface type on which RIP is running
- Example

```
host1#show ip rip brief
IP Address Tx  Rx  Auth  Met AccList O/I  RtMap Status Intf
10.2.1.32  1   1,2 none  1   no/no   no   up    fastEthernet0/0
10.10.1.2  1   1,2 none  1   no/no   no   up    serial5:1/1:1
```

show ip rip database

- Use to display the route entries in the RIP routing table.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Specify the **active** keyword to limit the display to active routes learned via RIP updates.
- Specify the **inactive** keyword to limit the display to routes that the router will discard in the immediate future.
- Field descriptions
  - Prefix—IP address prefix
  - Length—Prefix length
  - ttl—(Time to live) Indicates how many seconds the specific route remains in the routing table. If an entry reaches 0, it is removed from the routing table.

- Met—Metric that RIP uses to rate the value of different routes (hop count). The hop count is the number of routers that can be traversed in a route.
- Next Hop—Next IP address where a packet is sent. A value of zero in this field indicates that the next address the packet should be sent to is the router that originally sent the RIP message.
- Intf—Interface that the route has learned

- Example

```
host1#show ip rip database
Prefix/Length: ttl  Met:  Next Hop       Intf:
3.0.0.0/8      0    1     72.30.100.2    tm2/1.100
9.20.0.0/17    0    2     172.30.100.1   tm2/1.100
10.2.1.0/24    0    2     172.30.100.1   tm2/1.100
```

show ip rip network

- Use to display the networks associated with the RIP routing process.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Field descriptions
  - network—IP address of a network on which RIP is running
  - netmask—Network mask applied to the network address
- Example

```
host1#show ip rip network
Network         netmask
 10.2.1.0       255.255.255.0
 172.30.100.0   255.255.255.0
 172.30.200.0   255.255.255.0
```

show ip rip peer

- Use to display limited information about each RIP neighbor.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Field descriptions
  - Time since last update received—Time in seconds since an update was received from this peer
  - Peer version—Version of IS-IS running on the peer
  - Bad packets received—Number of bad packets received from the peer
  - Bad routes received—Number of bad routes received from the peer
  - BFD—State of the BFD session with the peer, Up or Down
- Example

```
host1#show ip rip peer
192.168.1.102
   Time since last update received = 24
   Peer version = 1
   Bad packet received = 0
   Bad routes received = 0
   BFD Up
```

```
192.168.1.151
   Time since last update received = 24
   Peer version = 1
   Bad packet received = 0
   Bad routes received = 0
   BFD Down

192.168.1.250
   Time since last update received = 7
   Peer version = 2
   Bad packet received = 0
   Bad routes received = 0
   BFD Up
```

show ip rip statistics

- Use to display global and session statistics counters for RIP. If you specify an IP address, statistics for that interface are displayed in addition to the global RIP statistics.
- Specify **vrf** *vrfName* to limit the display to a specific VRF.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown. You must use the **baseline ip rip** command to set a baseline.
- Field descriptions
  - Number of route changes—Number of times the router has been told to route changes by its peers
  - Number of route queries—Number of times the router has received route requests from other routers
  - Received bad packets—Number of bad packets received from the peer
  - Received bad routes—Number of bad routes received from the peer
  - Triggered updates sent—Number of triggered updates sent; triggered updates are sent before the entire RIP routing table is sent; triggered by events such as adding a new RIP route or redistribution
  - Received updates—Number of updates received
- Example 1

```
host1#show ip rip statistics
 Number of route changes = 23
 Number of route queries = 0
```

- Example 2

```
host1#show ip rip statistics 10.2.1.32
Number of route changes = 901
Number of route queries = 0

fastEthernet 0/0, 10.2.1.32
  Received bad packet = 0
  Received bad routes = 0
  Triggered updates sent = 2
  Received updates = 41
```

show ip rip summary-address

- Use to display the specified summary address or all summary addresses for RIP.
- Field descriptions
  - Summary Address—Address summarizing RIP routes
  - Mask—Network mask specified in the **ip summary-address** command to identify which routes to summarize
  - Metric—Metric advertised with the summary RIP prefix
- Example

```
host1#show ip rip summary-address
Summary Address Mask            Metric
 4.3.0.0         255.255.0.0     3
 4.4.0.0         255.255.0.0     5
```

Chapter 5

Configuring OSPF

This chapter provides information for configuring the Open Shortest Path First (OSPF) routing protocol on your E-series router; it contains the following sections:

Overview

OSPF is an interior gateway protocol (IGP) that runs within a single autonomous system (AS). Exterior gateway protocols (EGPs), such as Border Gateway Protocol (BGP), exchange routing information between ASs.

OSPF is a link-state routing protocol, similar to the Intermediate System–to–Intermediate System (IS-IS) routing protocol. It advertises the states of its local network links. This link advertisement distinguishes OSPF from some IGPs, such as Routing Information Protocol (RIP). A distance vector protocol, such as RIP, advertises the distances (that is, the number of hops) to each known destination within the network.

Each participating OSPF router within the AS has an identical database describing the AS's topology. Each individual piece of this database is a particular router's local state. From this database, OSPF calculates a routing table by constructing a shortest-path tree.

OSPF learns the best routes to reachable destinations. It can quickly detect changes in the topology of an AS and, after a short convergence period, calculate new loop-free routes. This protocol has been designed expressly for the TCP/IP Internet environment, including explicit support for classless interdomain routing (CIDR) and the tagging of externally derived routing information.

This chapter provides direction for customizing basic OSPF settings if you need to do so. For detailed information about the OSPF commands, see the *JUNOSe Command Reference Guide N to Z*.

OSPF Terms

Table 10 defines commonly used OSPF terms.

Table 10: OSPF-Related Terms

Term	Meaning
adjacency	The relationship between selected neighboring routers for exchanging routing information. Not every pair of neighboring routers is adjacent.
area	A collection of network segments interconnected by routers. It is a region in an OSPF routing domain.
area border router (ABR)	A router that sits on the edge of an OSPF area and routes link-state advertisements (LSAs) between areas.
area ID	A unique number that identifies an area. Typically, formatted as an IP address.

Table 10: OSPF-Related Terms (continued)

Term	Meaning
authentication	A process whereby a user or data source proves that it is what it claims to be.
authentication type	The method by which authentication is achieved—null (or none), simple, or MD5. For example, simple authentication requires a 64-bit password in each OSPF packet.
autonomous system (AS)	A set of networks or IP prefixes within a single routing policy domain.
autonomous system boundary router (AS boundary router)	An OSPF router that redistributes routing information from other routing protocol sources.
classless interdomain routing (CIDR)	An addressing method that replaces the traditional class structure of IP addresses. In CIDR, the boundary between the network and host portions of an IP address can be on any bit boundary. CIDR addresses have no class restrictions, enabling more efficient used of the IP address space. CIDR addresses are represented by a prefix and a notation that indicates the IP address and mask; for example, 10.12.8.3/16.
designated router	A designated device (OSPF router) with which other routers form adjacencies, reducing the number of adjacencies required on a broadcast or NBMA network.
domain	A collection of routers that use a common interior gateway protocol.
flooding	The distribution and synchronization of the link-state database between OSPF routers.
hello protocol	A protocol that establishes and maintains neighbor relationships and that communication between neighbors is bidirectional. The hello protocol also dynamically discovers neighboring routers on broadcast or point-to-point networks.
interior gateway protocol (IGP)	A routing protocol that routers within an AS use to exchange information.
link-state advertisement (LSA)	A unit of data that describes the local state of a router or network. LSAs are flooded throughout their respective flooding domains. For example, router LSAs are flooded within the area to which the router belongs, summary LSAs are flooded to other areas through the backbone, and external LSAs are flooded throughout the OSPF domain.

Table 10: OSPF-Related Terms (continued)

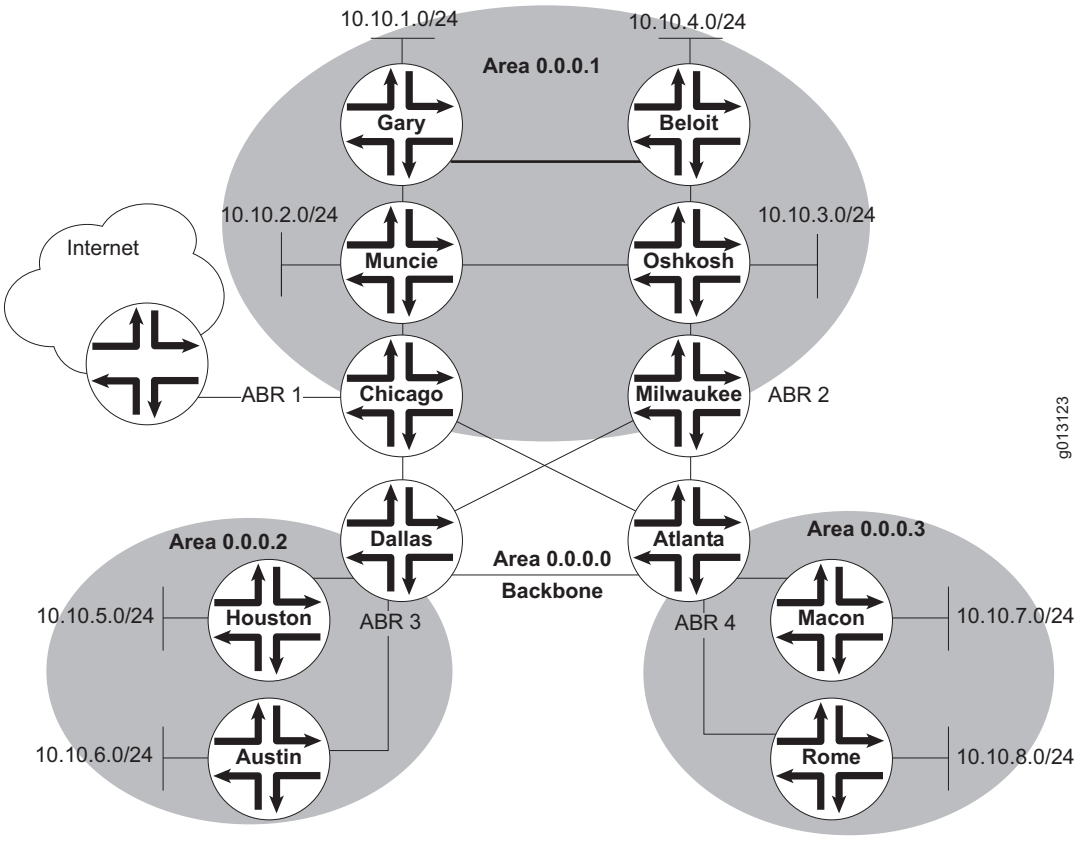
Term	Meaning
LSA types	OSPF LSAs are categorized into the following types: ■ Type 1—LSAs generated by an OSPF router for each area that it belongs to. Type 1 LSAs are flooded to only a single area. These LSAs carry information about directly connected links. Also known as router LSA. ■ Type 2—LSAs generated by an OSPF designated router to describe the set of routers in a network. Type 2 LSAs are flooded to the area that contains that network. Also known as network LSA. ■ Type 3—LSAs generated by an ABR to describe inter-area routes to networks outside of that area and internal to the AS; used for route summarization. Also known as inter-area prefix LSA. ■ Type 4—LSAs generated by an ABR to describe inter-area routes to ASBRs outside of that area and internal to the AS; used for route summarization. Also known as inter-area router LSA. ■ Type 5—LSAs generated by an ASBR to describe links that are external to the AS. Type 5 LSAs are reflooded from other protocols into OSPF, and are flooded by OSPF throughout the routing domain to all area types other than stub areas. OSPF sets the forwarding address for a type 5 LSA when the next hop is directly connected to the OSPF interface. Also known as AS-external LSA. ■ Type 6—Not supported. ■ Type 7—LSAs generated by an ASBR to describe routes that are external to an NSSA. Type 7 LSAs are flooded only to NSSAs. ■ Type 8—Not supported. ■ Type 9—Opaque LSA with a link-local scope. Type 9 LSAs are not flooded beyond the local network (local link). ■ Type 10—Opaque LSA with an area-local scope. Type-10 LSAs are not flooded beyond the borders of their associated area. ■ Type 11—Opaque LSA flooded throughout the AS. Type 11 LSAs are flooded throughout all transit areas, are not flooded into stub areas from the backbone, and are not originated by routers into their connected stub areas. Any type 11 LSA received in a stub area from a neighboring router within the stub area is rejected. ■ Link LSA—OSPFv3 LSA that Provides the router's link-local address to all other routers attached to the link; informs other routers attached to the link of a list of IPv6 prefixes to associate with the link; enables the router to assert a collection of options bits in the network LSA to be originated for the link ■ Intra-area prefix LSA—OSPFv3 LSA that associates a list of IPv6 address prefixes with a transit network link by referencing a network LSA, or associates a list of IPv6 address prefixes with a router by referencing a router LSA. The intra-area prefix LSA includes the IPv6 prefix information that OSPFv2 includes in type 1 and type 2 LSAs.
neighboring routers	Routers that have interfaces to a common network.
nonbroadcast network	A network that has no broadcast capability but supports more than two routers.
Not-so-stubby area (NSSA)	Similar to a stub area, but can also import selected external LSAs.
router ID	A 32-bit number that uniquely identifies a router within an AS; for example, 10.10.1.5.

Table 10: OSPF-Related Terms (continued)

Term	Meaning
stub area	An area that does not get flooded with external LSAs but does carry intra-area and interarea routes and a default route.
Totally stubby area	A stub area that also blocks type 3 summary LSAs from flowing into the area; however, type 3 LSAs carrying default route information alone are injected into the area.
virtual link	A logical link between two backbone routers for which the link tunnels through a nonbackbone area.

Figure 16 illustrates the topology of an OSPF routing domain.

Figure 16: OSPF Topology



Platform Considerations

For information about modules that support OSPF on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support OSPF.

For information about modules that support OSPF on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support OSPF.

References

If you need more information about the OSPF protocol, see the following documents:

- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—See the Release Notes corresponding to your software release for information about maximum values.
- OSPFv3 Graceful Restart—draft-ietf-ospf-ospfv3-graceful-restart-04.txt (November 2006 expiration)
- RFC 2328—OSPF Version 2 (April 1998)
- RFC 2370—The OSPF Opaque LSA Option (July 1998)
- RFC 2740—OSPF for IPv6
- RFC 3623—Graceful OSPF Restart (November 2003)
- RFC 3630—Traffic Engineering (TE) Extensions to OSPF Version 2 (September 2003)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

For information about the OSPF protocol working group, see http://www.ietf.org/html.charters/ospf-charter.html.

Features

The following sections provide brief descriptions of key OSPF features supported in our implementation of OSPF.

Intra-area, Interarea, and External Routes

You can split up an OSPF AS into areas. Doing this reduces the size of the link-state database (LSDB). Each OSPF area runs as a separate network and maintains its own LSDB. OSPF computes routes only to destinations within the area, and does not flood routes beyond the area boundaries.

Routing Priority

OSPF areas receive routes based on priority. Table 11 describes the routing priority.

Table 11: Routing Priority

Priority	Type	Description
1 (highest)	Intra-area	Intra-area routing. Refers to routing within a single OSPF area.
2	Interarea	Interarea routing. Refers to routing between OSPF areas within a single OSPF routing domain.
3	External	External type 1. Refers to routing from other protocols that can be imported into the OSPF domain and readvertised by OSPF as type 1 external. Type 1 metric is comparable to the link-state metric; the cost is equal to the sum of the internal costs plus the external cost.
4 (lowest)	External	External type 2. Refers to routing from other protocols that can be imported into the OSPF domain and readvertised by OSPF as type 2 external. Type 2 metric is much larger than the cost of any intra-AS path; the cost is equal to the external cost. This is the OSPF default.

If you use the **redistribute** command to import routes from other protocols or sources, the routes default to external type 2. You can specify a route map with the **redistribute** command to modify the type. Alternatively, you can use the **metric-type** keyword with the **redistribute** command to specify the type.

Virtual Links

Each OSPF area must be directly connected to the backbone area. The backbone is responsible for distributing routing information between nonbackbone areas. All routers in the backbone must be contiguous, but they need not be physically adjacent. You can configure backbone routers to be logically adjacent by creating OSPF virtual links.

Authentication

OSPF supports three modes of authentication:

- Null authentication—Implies that no authentication is in use.
- Simple password authentication—Requires a 64-bit unencrypted password in each OSPF packet.
- Cryptographic authentication—Uses a shared secret key that is configured on each router on a network. RFC 2328 defines the use of OSPF cryptographic authentication with the MD5 algorithm.

Opaque LSAs

OSPF opaque LSAs provide a generalized way of extending OSPF. The router generates opaque LSAs to carry traffic engineering information, accepts them from other routers, and floods them accordingly. OSPF uses the traffic engineering information to build a database from which paths can be computed for MPLS label-switched paths.

Route Leakage

Routes can be leaked into OSPF or from OSPF as follows:

- Route leakage into OSPF—When another routing protocol adds a new route to the routing table, or when a static route is added to the routing table, OSPF can be informed through the **redistribute** commands. When OSPF learns the new route, it floods the information into the routing domain by using external LSAs.
- Route leakage from OSPF—OSPF adds routing information to the routing table, which is used in forwarding IP packets.

Equal-Cost Multipath

OSPF inherently supports equal-cost multipath (ECMP). When building the shortest-path tree, OSPF calculates all paths of equal cost to a given destination. If equal-cost paths exist, OSPF inserts into the routing table the next hops for all equal-cost paths to a destination.

OSPF MIB

See the *JUNOSe Software CD,* shipped with your router, for complete information about the OSPF Management Information Base (MIB) supported by your router. The MIBs folder contains information about all supported standard and Juniper Networks E-series enterprise (proprietary) MIBs. OSPF does not act as a host within the router and therefore does not support the ospfIfMetric and ospfHost tables.

Interacting with Other Routing Protocols

OSPF interacts seamlessly with the following routing protocols:

- IS-IS—OSPF was developed originally from an early version of the IS-IS intradomain routing protocol. OSPF can import IS-IS routing information. See *Chapter 6, Configuring IS-IS*.
- RIP—E-series routers can simultaneously run OSPF and RIP. When doing so, OSPF routes are preferred over RIP. In general, use of the OSPF protocol is preferred because of its robustness, responsiveness, and decreased bandwidth requirements. See *Chapter 4, Configuring RIP*.
- BGP—The default expectation is that your routing environment is an AS running OSPF and exchanging BGP routes with other ASs. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 1, Configuring BGP Routing*.

Implementing OSPF for IPv6

OSPF version 3 (OSPFv3) specifies IPv6 support in the OSPF protocol. Compared with OSPF version 2, the fundamental mechanisms for OSPF remain unchanged. These mechanisms include the following:

- OSPF designated router/border designated router election
- OSPF adjacency maintenance
- OSPF interface states, events, and interface state machine
- OSPF flooding mechanism
- OSPF LSA management
- SPF calculation

Understanding the OSPFv3 Difference

OSPFv3 changes the way it describes the network topology. All addressing semantics have been removed from the LSA header and from router-LSAs and network-LSAs. These two LSAs now describe the topology of the routing domain in a network-protocol-independent manner (using interface identifiers and router identifiers). New LSAs have been added to distribute IPv6 address information and data required for next-hop resolution.

In addition to the obvious address and processing modifications to handle IPv6 addressing, changes in OSPFv3 include the following:

- Authentication-related information is removed from the OSPF packet headers. Instead, OSPFv3 uses an authentication header in IPv6.
- OSPFv3 requires that each OSPF interface attached to a link be assigned a link-local unicast address.
- The option field for hello packets, database description (DD) packets, and LSAs has been expanded from 8 bits to 24 bits. In addition, two new LSA types have been added—link LSAs and intra-area prefix LSAs.

- The LSA flooding scope is more explicit in OSPFv3 and now appears in the LS type field. The LS type field also encodes a specific action to take for unknown LS types, allowing OSPF to function with unknown LS types instead of simply discarding them.
- The flooding process is modified to manage unrecognized LSAs and the new LSA flooding scope.
- The route calculation has been updated to handle modifications in the LSA database.

Supported LSA Types

OSPFv3 supports the following LSA types:

- Router LSA—Describes link state and costs of router links to the area; flooded within an area only
- Network LSA—Originated by the designated router for every broadcast or nonbroadcast multiaccess (NBMA) link having two or more attached routers; lists all routers attached to the link
- Interarea prefix LSA—Known as the type-3 summary LSA in OSPFv2; describes a prefix external to the area, yet internal to the AS
- Interarea router LSA—Called type 4 summary-LSAs in OSPFv2; describes a path to a destination OSPF router (that is, an AS boundary router) that is external to the area, yet internal to the AS
- AS-external LSA—Describes a path to a prefix external to the AS
- Link LSA (new for OSPFv3)—Provides the router's link-local address to all other routers attached to the link; informs other routers attached to the link of a list of IPv6 prefixes to associate with the link; enables the router to assert a collection of options bits in the Network-LSA to be originated for the link
- Intra-area prefix LSA (new for OSPFv3)—Associates a list of IPv6 address prefixes with a transit network link by referencing a network LSA, or associates a list of IPv6 address prefixes with a router by referencing a router LSA

An LSA in OSPFv3 is still identified by its type, link-state ID, and the advertising router ID. However, the link-state ID (for all LSA types) no longer carries IP address information. Instead, the LSA carries either an arbitrarily assigned number or an interface ID.

The link-state ID always has a fixed length of 4 bytes. The LS type field is extended to 16 bits and encodes LSA flooding scope and specific actions to take when the router encounters unrecognized LS types.

An IPv6 address, if it is specified in an LSA, is represented by its prefix length, prefix options, and prefix address.

Unsupported OSPF Components

This release does not support the following OSPF components when implementing OSPF for IPv6:

- Virtual link
- Not-so-stubby-area (NSSA)
- Nonbroadcast multiaccess (NBMA)
- Remote neighbor
- Traffic engineering extensions
- SNMP traps
- Features specified in "OSPF as the PE/CE Protocol in BGP/MPLS IP VPNs" (draft-ietf-l3vpn-ospf-2547)

Configuration Tasks

Configuring OSPF requires careful coordination among a variety of routing devices:

- Routers internal to a single area
- Routers that link multiple areas within a single routing domain; these routers are called area border routers (ABRs)
- Routers that link multiple routing domains; these routers are called autonomous system boundary routers (AS boundary routers)

To minimally configure OSPF, you must:

1. Enable OSPF.
2. Configure and aggregate network ranges.
3. Create the router's OSPF network interfaces.
4. Define the OSPF areas attached to the router.

The following sections describe how to perform these tasks.

Starting OSPF

You enable OSPFv2 and OSPFv3 differently. When you enable OSPFv2 on your router, you can create either a range of OSPFv2 interfaces or a single OSPFv2 interface. When enabling OSPFv3, you create the OSPFv3 interface and assign the interface to an area.

Enabling OSPFv2

You can create OSPFv2 interfaces in the following ways:

- You can issue the **network area** command, which creates OSPF interfaces for all IP interfaces with IP addresses within the specified range.
- You can issue the **address area** command, which creates an OSPF interface in the specified area that sits on top of the IP interface at the given IP address (or on the unnumbered interface, if that is specified).

NOTE: Do not enable OSPF on any unidirectional interfaces (such as an MPLS tunnel), because it can never form an adjacency.

You can delete OSPFv2 interfaces in the following ways:

- You can issue the **no network area** command, which deletes all OSPF interfaces within the specified range.
- If the OSPF interface was created with the **address area** command, you can issue the **no address area** command to delete the specified interface.
- You can issue the **no ip address** command to delete the IP interface associated with the OSPF interface and also the OSPF interface itself.

NOTE: If an OSPF interface is configured on top of an IP interface and you delete the IP interface, the corresponding OSPF interface is also deleted. The previously configured network range, however, is not deleted. You must issue the **no network area** command to delete the range.

Enabling OSPFv3

NOTE: Before you can enable OSPFv3, you must specify an IPv6 license key. For additional information about configuring an IPv6 license key, see *Configuring an IPv6 License* on page 126.

OSPFv3 provides IPv6 support in the OSPF protocol. To enable OSPFv3:

1. Issue the **ipv6 router ospf** command, and specify a process ID.
2. Use the **router id** command to specify a router ID for OSPFv3.

   See *Specifying an OSPF Router ID* on page 244.
3. Issue the **ipv6 ospf area** command (in interface configuration mode) to create an OSPFv3 interface under an area ID.

You can delete OSPFv3 interfaces in the following ways:

- You can issue the **no ipv6 router ospf** command, which deletes OSPFv3.
- You can issue the **no ipv6 ospf area** command to remove the OSPF interface from a specific area.

Creating a Range of OSPF Interfaces

To create a range of OSPFv2 interfaces:

1. Create an OSPF routing process.
2. Create the range of IP addresses associated with the routing process and the corresponding OSPF interfaces.
3. Assign an area ID associated with each range of IP addresses.

Each router running OSPFv2 has a database describing a map of the routing domain. This map needs to be identical in all participating routers.

network area

- Use to configure a range of OSPFv2 interfaces and their related area.
- If the specified range matches one or more of the IP addresses configured for IP interfaces, one or more corresponding OSPF interfaces are created and placed in the specified area.
- Create address ranges that do not overlap; you can attach only the same range of interfaces to a single area.
- You cannot use this command for unnumbered interfaces.
- If the range specified by this command includes an address on an interface that is being referred to by unnumbered interfaces, all of the unnumbered interfaces begin trying to form adjacencies. If this behavior is not intended, you must reevaluate the interface assignment or the range specified by the command.

- Example 1—shows the creation of one OSPF interface in the backbone area

  host1(config-if)#**ip address 2.2.2.1 255.255.0.0**
  host1(config-if)#**ip address 2.2.1.1 255.255.0.0 secondary**
  host1(config)#**router ospf 2**
  host1(config-router)#**network 2.2.2.0 0.0.0.255 area 0**

- Example 2—shows the creation of two OSPF interfaces, one in the backbone area and one in a non-backbone area

  host1(config-if)#**ip address 2.2.2.1 255.255.255.0**
  host1(config-if)#**ip address 2.2.1.1 255.255.255.0 secondary**
  host1(config)#**router ospf 2**
  host1(config-router)#**network 2.2.2.0 0.0.0.255 area 0**
  host1(config-router)#**network 2.2.1.0 0.0.0.255 area 1**

  This sequence of commands creates two OSPF ranges (2.2.2.0/24 and 2.2.1.0/24), with each range belonging to a different area. Area 0 is configured for 2.2.2.0/24, and area 1 is configured for 2.2.1.0/24. This sequence also creates two OSPF interfaces: one in the backbone area (area 0) using IP address 2.2.2.1, the second in a nonbackbone area (area 1) using IP address 2.2.1.1. This command also creates the two areas if they do not already exist.

- Use the **no** version to delete OSPF interfaces, ranges, and areas.

NOTE: Until you activate the configured network range for summaries by issuing the **area range** command, the range is not active for summarization; the network range is summarized through area summaries—for ABRs only. (See *Aggregating OSPF Networks* on page 244.) The only range that is active by default if you do not issue the **area range** command is the network that matches the IP interface's network exactly. (In other words, by default the exact network of the IP interface is going to be summarized into other areas.)

ospf enable

- Use to enable OSPF on the router.
- OSPF is enabled by default.
- Example

  host1(config-router)#**ospf enable**

- The **no** version of this command is deprecated and may be removed in a future release. Use the **ospf shutdown** command to disable OSPF on the router.

router ospf
ipv6 router ospf

- Use to set an OSPF process ID.
- The process ID can be any positive integer in the range 1–65535.
- You must assign a unique ID for the OSPF routing process.
- From a virtual router context you can specify a VRF name (OSPFv2 only). Doing so changes the context to that of the specified VRF and remains so until you exit from the OSPFv2 router context.

- Example 1

  host1(config)#**router ospf 5**

- Example 2

  host1(config)#**ipv6 router ospf 5**

- Use the **no** version to end the designated OSPF routing process.

Creating a Single OSPFv2 Interface

To create a single OSPFv2 interface:

1. Create an OSPF routing process.
2. Create the OSPF interface associated with the IP interface at the specified address.

Each router running OSPF has a database describing a map of the routing domain. This map needs to be identical in all participating routers.

address area

- Use to create an interface in an area on which OSPFv2 runs, on top of the IP interface at the specified IP address.
- You can specify either an IP address or an unnumbered interface.
- Configures OSPFv2 with the default values. You can configure the interface with a nondefault value by using the other **address** commands. You must first issue the **address area** command before issuing any other **address** commands. See *Configuring OSPF Interfaces* on page 246 for more information.
- Example

  host1(config-router)#**address 10.10.32.100 area 0.0.0.0**

- Use the **no** version to delete the OSPFv2 interface.

ospf enable

- Use to enable OSPF on the router.
- OSPF is enabled by default.
- Example

  host1(config-router)#**ospf enable**

- The **no** version of this command is deprecated and might be removed in a future release. Use the **ospf shutdown** command to disable OSPF on the router.

router ospf

- Use to set an OSPF process ID.
- The process ID can be any positive integer in the range 1–65535.
- You must assign a unique ID for each OSPF routing process.
- Example

  host1(config)#**router ospf 5**

- Use the **no** version to end the designated OSPF routing process.

Specifying an OSPF Router ID

The router ID is typically derived by each router from its interface IP addresses. However, you can use the **router-id** command to specify a different router ID for OSPF.

NOTE: You must specify a router ID to enable OSPFv3.

router-id

- Use to specify a different IP address for the router to use as the OSPF router ID.
- Example

  host1(config-if)#**router-id 192.168.50.5**

- Use the **no** version to force OSPF to use the previous OSPF router ID behavior.

Aggregating OSPF Networks

You can aggregate OSPF networks at the border of an OSPF area by using the **area range** command. You can also aggregate OSPF networks when entering the border of the OSPF domain by using the **summary-address** command for IP routes and the **summary-prefix** command for IPv6 routes.

To create an area range:

1. Configure the interface's IP addresses using the **ip address** command.
2. Enable OSPF using the **router ospf** command.
3. Configure the network area with the **network area** command.
4. Configure the area range with the **area range** command.

area range

- Use to aggregate OSPF routes at an OSPF area border.
- Use only for ABRs.
- You can configure multiple instances of the **area range** command for a single OSPF area.

- By default, the range of configured networks is advertised in type 3 (summary) LSAs.
- Use the **advertise** keyword (IPv6 only) to specify advertisement of configured networks.
- Use the **do-not-advertise** keyword to prevent advertisement of configured networks.
- Use the **cost** keyword (IPv6 only) to define the cost value (0–16777215) for the specified range of networks.
- Use the command **no area** *area-id* (with no other keywords) to remove the specified area from the configuration.
- Use the **summary-address** or **summary-prefix** command to summarize external routes being redistributed into OSPF.
- Example

  host1(config-if)#**ip address 2.2.10.1 255.255.255.0**
  host1(config-if)#**ip address 2.2.11.1 255.255.255.0 secondary**
  host1(config)#**router ospf 2**
  host1(config-router)#**network 2.2.0.0 0.0.255.255 area 0**

  At this point, the OSPF process is configured with two OSPF interfaces. If your router is an ABR, two networks must be summarized: 2.2.10.0/24 and 2.2.11.0/24.

  host1(config-router)#**area 0 range 2.2.0.0 255.255.0.0**

  After you enter this **area range** command, only the aggregated range 2.2.0.0/16 is going to be summarized.

- Use the **no** version to disable the aggregation of routes at the OSPF area border.

summary-address
summary-prefix

- Use to aggregate external routes at the border of the OSPF routing domain.
- Use the **summary-address** command for IP routes. Use the **summary-prefix** command for IPv6 routes.
- Use only for AS boundary routers.
- The AS boundary router advertises one external route as an aggregate for all redistributed routes that are covered by the address.
- For OSPF, these commands summarize only routes from other routing protocols that are being redistributed into OSPF.
- With these commands, you can reduce the load of advertising many OSPF external routes by specifying a range that includes some (or all) of these external routes.
- Example

  host1(config-router)#**summary-address 10.1.0.0 255.255.0.0**

- Use the **area range** command for route summarization between OSPF areas.
- Use the **no** version to restore the default.

Configuring OSPF Interfaces

You can configure OSPF attributes for either a single OSPF network by using the **address** commands, or for all OSPF networks on a particular media interface by using the **ip ospf** commands.

The size of the OSPF maximum transmission unit (MTU) is negotiated rather than configured. OSPF database description exchange uses the interface MTU to signal the largest OSPF MTU that can be sent over an OSPF interface without fragmentation.

Configuring OSPF attributes for OSPF networks includes setting the following:

- Cost
- Dead interval
- Hello interval
- Router priority
- Retransmit interval
- Transmit delay

NOTE: Before using the **address** or **ip ospf** commands, see *Precedence of Commands* on page 253 for information about the relationship between these commands.

address Commands

You can use the **address area** command to create a new OSPF interface. Use the other **address** commands to configure parameters for OSPF interfaces that already exist.

The **address** commands configure OSPF attributes for a single OSPF network. The **ip ospf** commands configure OSPF attributes for all OSPF networks in the given interface context—for example, in a multinet environment where multiple IP networks sit on top of an Ethernet interface.

NOTE: You must first issue the **address area** command before issuing any other **address** command.

address area

- Use to create a new OSPF interface and configure the area ID.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 10.12.10.2 area 3**

- You must first issue the **address area** command before issuing any other **address** commands.
- Use the **no** version to delete the area ID from the specified interface.

address cost

- Use to specify the cost metric for the interface. The cost is used in calculating the SPF routing table and can be in the range 0–65535.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address unnumbered atm 4/0.1 area 3**
  host1(config-router)#**address unnumbered atm 4/0.1 cost 50**

- Use the **no** version to reset the path cost to the default value, 1.

address dead-interval

- Use to specify the time period for the router's neighbors to wait without seeing hello packets from the router before they declare the router to be down.
- The dead interval can be in the range 0–2147483647 seconds, and is advertised by the router's hello packets.
- For the OSPF routers to become adjacent, the dead interval must be identical on each router.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 192.168.10.32 area 6**
  host1(config-router)#**address 192.168.10.32 dead-interval 60**

- Use the **no** version to reset the dead interval to the default value, 40 seconds.

address hello-interval

- Use to specify the interval between hello packets that the router sends on the interface.
- The hello interval can be in the range 1–65535 seconds.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 192.168.1.1 area 5**
  host1(config-router)#**address 192.168.1.1 hello-interval 25**

- Use the **no** version to reset the hello interval to the default value, 10 seconds.

address passive-interface

- Use to disable the transmission of routing updates on the interface, meaning that OSPF routing information is neither sent by nor received through the interface.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 192.168.100.20 area 5**
  host1(config-router)#**address 192.168.100.20 passive-interface**

- Use the **no** version to reenable the transmission of routing updates.

address priority

- Use to specify the router priority, an 8-bit number in the range 1–255. Used in determining the designated router for the particular network.
- Applies only to nonbroadcast multiaccess (NBMA) networks. Every broadcast and NBMA network has a designated router.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address unnumbered loopback 0 area 6**
  host1(config-router)#**address unnumbered loopback 0 priority**

- Use the **no** version to restore the default value, 1.

address retransmit-interval

- Use to specify the time between LSA retransmissions for the interface when an acknowledgment for the LSA is not received.
- Specify an interval in the range 0–3600 seconds; the default value is 5.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 192.168.10.200 area 6**
  host1(config-router)#**address 192.168.10.200 retransmit-interval 500**

- Use the **no** version to restore the default value, 5 seconds.

address transmit-delay

- Use to specify the estimated time it takes to transmit a link-state update packet on the interface.
- Specify an interval in the range 0–3600 seconds; the default value is 1.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 10.100.25.38 area 7**
  host1(config-router)#**address 10.100.25.38 transmit-delay 30**

- Use the **no** version to restore the default value, 1 second.

ip ospf and ipv6 ospf Commands

The **ip ospf** commands have two effects on interface configuration. These effects apply to all **ip ospf** commands:

- Configuration per logical IP interface (for example, Fast Ethernet 0/1.3 or ATM 5/0.1):

  The **ip ospf** command configures the specified OSPF parameters for all networks configured on the given IP interface—for example, all multinetted addresses on an interface.

  The **no** version of the command resets the specified parameters to *unspecified*.

  If the **no** version of the command takes effect for a specified IP interface, there is no default value for the specified parameters. The parameter is set back to unspecified values. However, the value of the specified parameter for the OSPF interface is set back to the default value or the value previously specified by the **address** command.

NOTE: The **ip ospf** commands configure OSPF attributes for all OSPF networks in the given interface context—for example, in a multinet environment where multiple IP networks sit on top of an Ethernet interface. The **address** commands configure OSPF attributes for a single OSPF interface.

- Configuration per OSPF interface:

  The **ip ospf** command configures the specified OSPF parameters for *each* OSPF interface that sits on top of the IP interface.

  The **no** version of the command restores the specified parameters to the default values.

NOTE: We recommend using **address** commands to set attributes of OSPF interfaces created using the **address area** command.

ipv6 ospf area

- Use to create an OSPFv3 interface under the specified area ID or move the OSPFv3 interface from its current area to the specified area.
- Specify an optional process ID in the range 1–65535.
- Example

  host1(config)#**interface fastethernet 0/0**
  host1(config-if)#**ipv6 ospf area 50**

- Use the **no** version to remove this interface from the specified area.

ip ospf cost
ipv6 ospf cost

- Use to configure the cost of sending a packet on the network.
- Cost is a metric value in the range 0–65535; the default value is 1.
- The router LSA advertises the link-state metric as the link cost.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.
- Example 1

  host1(config)#**interface fastethernet 0/0**
  host1(config-if)#**ip ospf cost 50**

- Example 2

  host1(config)#**interface fastethernet 0/0**
  host1(config-if)#**ipv6 ospf cost 50**

- Use the **no** version to reset the path cost to the default value, 1.

ip ospf dead-interval
ipv6 ospf dead-interval

- Use to configure the interval since the last hello packet was seen.
- Specify an interval in the range 0–2147483647 seconds; the default value is 40 seconds.
- For the OSPF routers to become adjacent, the dead interval must be identical on each router.
- The router's hello packets advertise this interval.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.
- Example 1

  host1(config-if)#**ip ospf dead-interval 60**

- Example 2

  host1(config-if)#**ipv6 ospf dead-interval 60**

- Use the **no** version to restore the default value, 40 seconds.

ip ospf hello-interval
ipv6 ospf hello-interval

- Use to configure the interval between hello packets.
- Specify an interval in the range 1–65535 seconds; the default value is 10 seconds.
- For the OSPF routers to become adjacent, the hello interval must be identical on each router.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.

- Example 1

  host1(config-if)#**ip ospf hello-interval 8**

- Example 2

  host1(config-if)#**ipv6 ospf hello-interval 8**

- Use the **no** version to restore the default value, 10 seconds.

ipv6 ospf mtu-ignore

- Use to specify that the interface disregard the MTU size contained in the data description packet.
- When enabled, the interface accepts data description packets from its neighbor even if it has a different MTU size (the MTU size must be less than 18000).
- Specify an optional process ID in the range 1–65535.
- Example

  host1(config-if)#**ipv6 ospf mtu-ignore**

- Use the **no** version to reset the default: that the neighbor MTU size must match the MTU size of the OSPFv3 interface from which the packet is received.

ipv6 ospf network

- Use to configure the OSPF network type for an interface.
- Specify a network type (broadcast or point-to-point) for the interface.
- Example

  host1(config)#**interface fastethernet 0/0**
  host1(config-if)#**ipv6 ospf network broadcast**

- Use the **no** version to revert the network type to the default for the interface.

ip ospf priority
ipv6 ospf priority

- Use to configure the router's priority.
- Select a priority level in the range 0–255; the default value is 1.
- This setting determines the designated router for the particular network.
- A router whose priority is set to 0 cannot be a designated router.
- Configure priority only for interfaces to multiaccess networks.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.

- Example 1

  host1(config-if)#**ip ospf priority 2**

- Example 2

  host1(config-if)#**ipv6 ospf priority 2**

- Use the **no** version to restore the default value, 1.

ip ospf retransmit-interval
ipv6 ospf retransmit-interval

- Use to configure the time interval between retransmission of an LSA.
- Specify an interval in the range 0–3600 seconds; the default value is 5 seconds.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.
- Example 1

  host1(config-if)#**ip ospf retransmit-interval 10**

- Example 2

  host1(config-if)#**ipv6 ospf retransmit-interval 10**

- Use the **no** version to return to the default value, 5 seconds.

ip ospf transmit-delay
ipv6 ospf transmit-delay

- Use to configure the time it takes to transmit a link-state update on the interface.
- This is the time between transmissions of LSAs.
- Specify an interval in the rang 0–3600 seconds; the default value is 1 second.
- In setting the time, consider the interface's transmission and propagation delays.
- For the IPv6 command, you can specify an optional process ID in the range 1–65535.
- Example 1

  host1(config-if)#**ip ospf transmit-delay 4**

- Example 2

  host1(config-if)#**ipv6 ospf transmit-delay 4**

- Use the **no** version to return to the default value, 1 second.

Comparison Example

In the following example you configure a range of OSPF interfaces with the **network area** command.

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#ip address 1.1.1.1 255.255.255.0
host1(config-if)#ip address 2.2.2.2 255.255.255.0 secondary
host1(config-if)#exit
host1(config)#router ospf 1
host1(config-router)#network 1.1.1.0 0.0.0.255 area 0
host1(config-router)#network 2.2.2.0 0.0.0.255 area 0
```

If you want to specify the cost, you can do so for both interfaces simultaneously.

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#ip ospf cost 30
```

You can use **address** commands to create a third OSPF interface over the Ethernet interface. When you specify a cost, you set it for only that interface.

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#ip address 3.3.3.3 255.255.255.0 secondary
host1(config-if)#exit
host1(config)#router ospf 1
host1(config-router)#address 3.3.3.3 area 0
host1(config-router)#address 3.3.3.3 cost 25
```

Precedence of Commands

For a single OSPF interface, when you modify the same OSPF attribute by issuing both the **ip ospf** command and the **address** command, the value configured with the **address** command takes precedence. In other words, the most specific command for a single OSPF interface takes precedence.

Consider the following example. Suppose you have a numbered IP interface with an IP address of 10.10.1.1/24 sitting on top of Fast Ethernet interface 0/0. Configure a single OSPF interface on top of the IP interface.

```
host1(config)#router ospf 100
host1(router-config)#address 10.10.1.1 area 0
```

The default cost for this OSPF interface is 10. Change the cost for this OSPF interface by using the **address cost** command.

```
host1(router-config)#address 10.10.1.1 cost 45
```

The cost for OSPF interface 10.10.1.1 is now 45.

Now use the **ip ospf cost** command to change the cost for this OSPF interface.

```
host1(config)#int fastEthernet 0/0
host1(config-if)#ip ospf cost 23
```

The cost of OSPF interface 10.10.1.1 does *not* change. The previously issued **address cost** command is more specific for the interface and takes precedence over the **ip ospf cost** command. You must use the **address cost** command if you want to change the cost again.

```
host1(router-config)#address 10.10.1.1 cost 23
```

Configuring OSPF Areas

You can divide your OSPF routing domain into OSPF areas. Dividing into areas provides the following benefits:

- Reduces resource demands placed on routers and links
- Reduces the router CPU usage by the OSPF routing calculation
- Reduces the amount of memory used for link-state databases
- Hides subnets within areas from the rest of the routing domain
- Increases routing security within the area

You must attach each area in your routing domain to an area called the backbone area (0.0.0.0).

Disadvantages of using OSPF areas include the following:

- Areas hide information, which can result in less-than-optimal data paths.
- Creating areas complicates the task of configuring OSPF routing domains.

You can optionally define an area to be a stub area, totally stubby area, or a not-so-stubby area. You can configure virtual links for areas that are not directly connected to a backbone area.

area default-cost

- Use to configure the cost for the default summary route sent into a stub area.
- Cost is a metric value in the range 1–65535; the default value is 1.
- Use only on an ABR attached to a stub area.
- Provides the metric for the summary default route that the ABR generates into the stub area.
- Example

  ```
  host1(config-router)#area 47.0.0.0 default-cost 1
  ```

- Use the **no** version to remove the configured default route cost.

area nssa

- Use to configure the area as an NSSA.
- You must configure each router in a stub area as belonging to the stub area.
- An NSSA is like a stub area, but it can also import external AS routes in a limited way.
- To cause NSSA border routers to generate a type 7 default LSA in the OSPF database if there is a default route in the routing table, you must specify the **default-information-originate** option.
- You can specify a metric cost, metric type, or a route map to be applied to the generated type 7 default LSAs.
- Use the **no-summary** keyword to create a "totally stubby area" and restrict type 3 summary LSAs from flowing into the area. However, type 3 default-route LSAs can continue to flow into the area and a type 3 default-route LSA is advertised into the NSSA.

NOTE: We recommend that you do not use the **default-information-originate** keyword with the **no-summary** keyword for an NSSA.

- Example

  host1(config-router)#**area 35.0.0.0 nssa**

- Use the **no** version to remove the NSSA designation from the area, to stop the generation of type 7 default LSAs, to reinitiate type 3 summary LSAs into the area (with the **no-summary** keyword), or to stop the application of the specified metric cost, metric type, or a route map to the type 7 default LSAs.

area stub

- Use to configure a stub area. Stub areas do not get flooded with external LSAs but do carry a default route, intra-area routes, and interarea routes. The lack of flooding in stub areas reduces the size of the OSPF database for the area and decreases memory usage for external routers in the stub area.
- You must configure each router in a stub area as belonging to the stub area.
- You cannot configure virtual links across a stub area.
- Stub areas cannot contain AS boundary routers.
- Use the **no-summary** keyword to create a "totally stubby area" and restrict type 3 summary LSAs from entering the stub area. However, type 3 default-route LSAs can continue to flow into the area.
- Example

  host1(config-router)#**area 47.0.0.0 stub**

- Use the **no** version to disable this function.

area virtual-link

- Use to configure an OSPF virtual link.
- A virtual link is used for areas that do not have a direct connection to the backbone area.
- To have configured virtual links, the router itself must be an ABR.
- Virtual links are identified by the router ID of the other endpoint, which is also an ABR.
- The two endpoint routers must be attached to a common area, called the virtual link's transit area.
- Virtual links are part of the backbone and behave as if they were unnumbered point-to-point networks between the two routers.
- A virtual link uses the intra-area routing of its transit area to forward packets.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.8.4.2**

- Use the **no** version to remove an OSPF virtual link.

area virtual-link dead-interval

- Use to set the time in seconds to wait before declaring a neighbor down after not receiving packets from that neighbor.
- Specify an interval in the range 0–2147483647 seconds; the default value is 40 seconds.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.8.4.2 dead-interval 10**

- Use the **no** version to remove the virtual link's dead interval.

area virtual-link hello-interval

- Use to configure the hello interval on an OSPF virtual link.
- Specify an interval in the range 1–65535 seconds; the default value is 10 seconds.
- The hello interval is the time between the transmission of hello packets.
- The hello interval must be the same for all routers attached to a common network.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.8.4.2 hello-interval 10**

- Use the **no** version to remove the virtual link's hello interval.

area virtual-link retransmit-interval

- Use to configure the retransmission interval on an OSPF virtual link.
- The retransmit interval is the time between retransmissions of link-state advertisements for adjacencies belonging to the interface.
- Specify an interval in the range 0–3600 seconds; the default value is 5 seconds.
- Set the value greater than the expected round-trip delay.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.8.4.2 retransmit-interval 6**

- Use the **no** version to remove the interface's retransmit interval.

area virtual-link transmit-delay

- Use to configure the estimated time it takes to transmit a link-state update packet on the virtual link.
- Specify an interval in the range 0–3600 seconds; the default value is 1 second.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.8.4.2 transmit-delay 1**

- Use the **no** version to remove the interface's transmit delay.

automatic-virtual-link

- Use to enable an automatic virtual link configuration.
- If this feature is enabled, then backbone connectivity is ensured by the automatic creation of a virtual link between this backbone router that has an interface to a common nonbackbone area and other backbone routers that have interfaces to a common nonbackbone area.
- Example

  host1(config-router)#**automatic-virtual-link**

- Use the **no** version to disable an automatic virtual link.

no area

- Use to remove the specified area only if no OSPF interfaces are configured in the area.
- Example

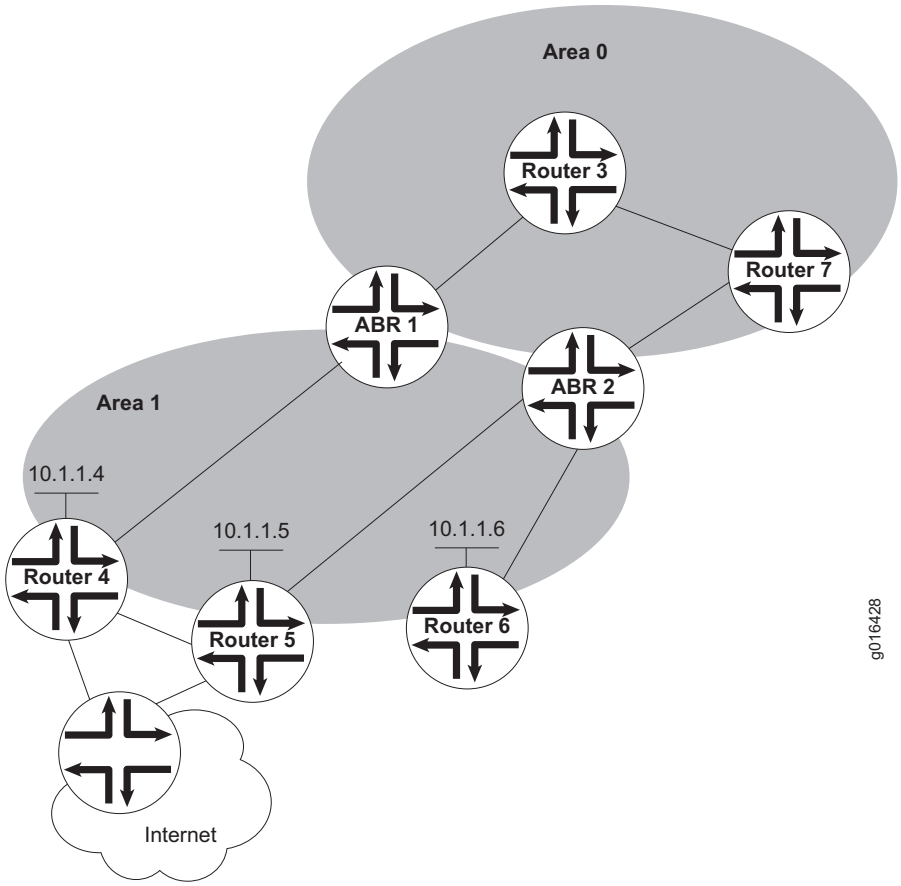
  host1(config-router)#**no area 47.0.0.0**

- There is no affirmative version of this command; there is only a **no** version.

Optimizing the Cost to Reach a Range of OSPF Routers Within an Area

OSPF automatically calculates a cost for an area based on the individual costs from an area border router to each OSPF router within that area. The highest individual cost is advertised by the area border router as the aggregate cost for routers in an adjacent area to reach any router within the first area. Consider the topology shown in Figure 17 on page 258.

Figure 17: Optimizing OSPF Area Aggregate Costs



In this example, the router IDs of the OSPF routers in area 1 are announced by OSPF into area 0. ABR 1 and ABR 2 aggregates the 10.1.1.x networks in area 1 at the border. Each individual OSPF link has a cost of 1.

ABR 1 calculates the following costs:

- A cost of 5 to reach Router 6:
  ABR 1--> Router 4--> Router 5--> ABR 2--> Router 6
- A cost of 3 to reach Router 5:
  ABR 1--> Router 4--> Router 5
- A cost of 2 to reach Router 4:
  ABR 1-> Router 4

The highest individual cost is 5. ABR 1 subsequently advertises a cost of 5 for the aggregate 10.1.1.0 to be announced into area 0.

ABR 2 calculates the following costs:

- A cost of 2 to reach Router 6:
  ABR 2--> Router 6

- A cost of 2 to reach Router 5:
  ABR 2--> Router 5

- A cost of 3 to reach Router 4:
  ABR 2--> Router 5--> Router 4

The highest individual cost is 3. ABR 2 subsequently calculates a cost of 3 for the aggregate 10.1.1.0 to be announced into area 0.

When Router 3 sends traffic to Router 4, it routes the traffic via ABR 2 because ABR 2 advertises a lower cost than does ABR 1. However, this path is not optimal, because the traffic must traverse Router 3--> Router 7--> ABR 2--> Router 5--> Router 4. The path through ABR 1, Router 3--> ABR 1--> Router 4 is a better path, even though ABR 1 advertised a higher aggregate cost.

You can avoid this kind of suboptimal routing by manually configuring a cost for the aggregate. The summary LSA then announces the configured cost instead of the automatically calculated cost. Use the **cost** keyword with the **area range** command to specify a cost for a range of OSPF networks aggregated at an area boundary.

Configuring Authentication

The router supports the following authentication capabilities:

- Null authentication
- Simple password authentication
- MD5 authentication

The MD5 algorithm takes as input a message of arbitrary length and produces a 128-bit *fingerprint* or *message digest* of the input. MD5 is used to create digital signatures. It is a one-way *hash* function, meaning that it takes a message and converts it into a fixed string of digits, called a message digest.

When using a one-way hash function, you can compare a calculated message digest with the message digest that is decrypted by using a public key (password). The key verifies that the message has not been tampered with. This comparison process is called a hashcheck.

NOTE: You must first issue the **address area** command before issuing any other **address** command.

Authentication Requirements

If you configure either simple password or MD5 authentication, the password or authentication key must be the same on both sides of an adjacency. When you change the password or key on one side of an established adjacency, you must also change it on the other side within the dead interval. Doing this enables a hello packet that has the latest authentication information to be sent before the dead interval expires. If the packet is not sent within the dead interval, the adjacency breaks down and is not reestablished until both sides of the adjacency have the same password or key.

address authentication-key

- Use to assign a password used by neighboring routers for OSPF simple password authentication.
- The interface can have an IP address, or it can be unnumbered.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- The password, or key, is a character string up to 8 characters long.
- Example

  host1(config-router)#**address 10.12.10.2 authentication-key 9rdf7**

- Use the **no** version to delete the password from the specified interface.

address authentication message-digest

- Use to specify that MD5 authentication is used for the OSPF interface.
- You must configure the MD5 key ID and password with the **address message-digest-key md5** command.
- Switching between authentication types does not delete a configured MD5 key ID or password; only using the **no** version of that configuration command can delete the MD5 key ID and password.
- Example

  host1(config-router)#**address 10.12.10.2 authentication message-digest**

- Use the **no** version to set authentication for the interface to none without removing any configured MD5 key. You can subsequently apply MD5 authentication to the interface without having to reconfigure the key.

address authentication-none

- Use to disable authentication on the interface.
- The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 192.168.10.32 authentication-none**

- The **no** version has no effect.

address message-digest-key md5

- Use to enable OSPF MD5 authentication and configure the MD5 key.
- The MD5 key is a character string up to 16 characters long. You must also specify a key identifier and whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Configures an interface already created, or creates a new OSPF interface and configures the MD5 key. The interface can have an IP address, or it can be unnumbered.
- Example

  host1(config-router)#**address 10.1.1.1 message-digest-key 1 md5 0 9mwk6gdr76**

- Use the **no** version to delete the MD5 key.

area virtual-link authentication-key

- Use to configure a simple password for a virtual link.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- The password can be up to eight characters long.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.3.4.5 authentication-key sadsa29c**

- Use the **no** version to remove the password.

area virtual-link authentication message-digest

- Use to specify that MD5 authentication is used for the particular virtual link.
- You must configure the MD5 key ID and password with the **area virtual-link message-digest-key md5** command.
- Switching between authentication types does not delete a configured MD5 key ID or password; only using the **no** version of that configuration command can delete the MD5 key ID and password.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.2.3.4 authentication message-digest**

- Use the **no** version to set authentication for the virtual link to none without removing any configured MD5 key. You can subsequently apply MD5 authentication to the virtual link without having to reconfigure the key.

area virtual-link authentication-none

- Use to specify that no authentication is used for the particular virtual link.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 27.2.3.4 authentication-none**

- The **no** version has no effect.

area virtual-link message-digest-key md5

- Use to enable MD5 authentication and to configure MD5 keys for virtual links.
- The MD5 key is a character string up to 16 characters long. You must also specify a key identifier and whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**area 27.0.0.0 virtual-link 327.3.4.5 message-digest-key 2 md5 rc45lsm2c**

- Use the **no** version to remove the password.

ip ospf authentication-key

- Use to configure a type 1 authentication (a simple password) on the interface.
- Neighboring OSPF routers use the password to access the router's interface.
- Use the same password on all neighboring routers on the same network.
- Use this password only when you enable authentication for the interface.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Use a password that is a continuous string up to 8 characters long.
- Example

  host1(config-if)#**ip ospf authentication-key yourpwd**

- Use the **no** version to remove the password on the interface.

ip ospf authentication message-digest

- Use to specify the authentication method for the interface as MD5.
- You must configure the MD5 key ID and password with the **ip ospf message-digest-key md5** command.
- Switching between authentication types does not delete a configured MD5 key ID or password; only using the **no** version of that configuration command can delete the MD5 key ID and password.
- Example

  host1(config-if)#**ip ospf authentication message-digest**

- Use the **no** version to set authentication for the interface to none without removing any configured MD5 key. You can subsequently apply MD5 authentication to the interface without having to reconfigure the key.

ip ospf authentication-none

- Use to specify that no authentication is used for the OSPF interface.
- Example

  host1(config-if)#**ip ospf authentication-none**

- The **no** version has no effect.

ip ospf message-digest-key md5

- Use to enable MD5 authentication on the OSPF interface and configure the MD5 key.

NOTE: If all the MD5 keys have been deleted, the authentication type is still MD5, but you need to configure MD5 keys.

- The MD5 key is a character string up to 16 characters long. You must also specify a key identifier and whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.

NOTE: To display the password only in encrypted text, use the **service password-encryption** command.

- Example

  host1(config-if)#**ip ospf message-digest-key 3 md5 0 tre987is**

- Use the **no** version to delete an MD5 key from the OSPF interface.

NOTE: To disable MD5 authentication for the interface, use the **ip ospf authentication-none** command.

Configuring the BFD Protocol for OSPF

The **ip ospf bfd-liveness-detection** and **ipv6 ospf bfd-liveness-detection** commands configure the Bidirectional Forwarding Detection (BFD) protocol for OSPFv2 and OSPFv3 (respectively). The BFD protocol uses control packets and shorter detection time limits to more rapidly detect failures in a network. Also, because they are adjustable, you can modify the BFD timers for more or less aggressive failure detection.

When you issue the **ip ospf bfd-liveness-detection** or **ipv6 ospf bfd-liveness-detection** command on an OSPF peer, the peer establishes BFD liveness detection with all BFD-enabled OSPF peers. When the local peer receives an update from a remote OSPF peer—if BFD is enabled and if the session is not already present—the local peer attempts to create a BFD session to the remote peer.

Each adjacent pair of peers negotiates an acceptable transmit interval for BFD packets. The negotiated value can be different on each peer. Each peer then calculates a BFD liveness detection interval. When a peer does not receive a BFD packet within the detection interval, it declares the BFD session to be down and purges all routes learned from the remote peer.

NOTE: Before the router can use the **ip ospf bfd-liveness-detection** or **ipv6 ospf bfd-liveness-detection** command, you must specify a BFD license key. To view an already configured license, use the **show license bfd** command.

For general information about configuring and monitoring the BFD protocol, see *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.

ip ospf bfd-liveness-detection
ipv6 ospf bfd-liveness-detection

- Use to enable BFD (bidirectional forwarding detection) and define BFD values to more quickly detect OSPFv2 or OSPFv3 data path failures.
- The peers in an OSPF adjacency use the configured values to negotiate the actual transmit intervals for BFD packets.
  - You can use the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-interval** keyword to specify the same value for both of those intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value. The default value is 300 milliseconds.
- You can use the **multiplier** keyword to specify the detection multiplier value. The calculated BFD liveness detection interval can be different on each peer. The multiplier value is roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. The default value is 3.
- For details on liveness detection negotiation, see *Negotiation of the BFD Liveness Detection Interval* section in *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.
- You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.
- Example 1 (OSPFv2)

  host1(config)#**ip ospf bfd-liveness-detection minimum-interval 800**

- Example 2 (OSPFv3)

  host1(config)#**ipv6 ospf bfd-liveness-detection minimum-interval 800**

- Use the **no** version to disable BFD on the OSPF interface.

Configuring Additional Parameters

The commands presented in this section include both OSPF-specific commands and routing protocol–independent commands that are not limited to OSPF. You can use these commands to perform the tasks listed in Table 12.

Table 12: Additional Configuration Tasks

Filter and apply policy to routes.	Set the maximum paths.
Set a baseline for statistics.	Enable automatic cost calculation.
Clear statistics for access lists, counters, redistributed routes, or processes.	Enable logs for OSPF neighbor changes.
Set the redistribution routes.	Set SPF hold time.
Set the distance for OSPF routes.	Set a default metric.
Administratively disable OSPF.	

access-list
route-map

- Use the **access-list** command to create a standard or extended access list.
- Use the **route-map** command to create a route map.
- For detailed information about configuring access lists and route maps, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.
- Example

  1. Configure three static routes.

     host1(config)#**ip route 20.20.20.0 255.255.255.0 192.168.1.0**
     host1(config)#**ip route 20.20.21.0 255.255.255.0 192.168.1.0**
     host1(config)#**ip route 20.21.0.0 255.255.255.0 192.168.1.0**

  2. Configure an access list with filters on routes 20.20.20.0/24 and 20.20.21.0/24.

     host1(config)#**access-list boston permit 20.20.0.0 0.0.255.255**

  3. Configure a route map that matches the previous access list and applies a metric type 1 (OSPF).

     host1(config)#**route-map boston**
     host1(config-route-map)#**match ip address boston**
     host1(config-route-map)#**set metric-type type-1**

  4. Configure redistribution of the static routes into OSPF with route map boston.

     host1(config)#**router ospf 2**
     host1(config-router)#**redistribute static route-map boston**

5. Use the **show ip ospf database** command to verify the effect of the redistribution (that the two static routes matching the route map are redistributed as external type 1).

```
host1#show ip ospf database
    OSPF Database
      Router Link States (Area 0.0.0.0)
Link ID         ADV Router      Age         Seq#        Checksum
192.168.1.250   192.168.1.250   3           0x80000006  0x39a1
192.168.254.7   192.168.254.7   220         0x80000169  0xd2b5
      Network Link States (Area 0.0.0.0)
Link ID         ADV Router      Age         Seq#        Checksum
192.168.1.214   192.168.254.7   220         0x80000001  0xe0f2
      AS External Link States
Link ID         ADV Router      Age         Seq#        Checksum
20.20.20.0      192.168.1.250   3           0x80000001  0x6045
20.20.21.0      192.168.1.250   3           0x80000001  0x554f
```

- Use the **no** version of the **access-list** command to remove the access list or the specified entry in the access list.
- Use the **no** version of the **route-map** command to remove an entry.

auto-cost reference-bandwidth
ospf auto-cost reference-bandwidth

- Use to calculate the OSPFv2 or OSPFv3 interface cost according to bandwidth.
- Sets the OSPF metric for an interface according to the bandwidth specified.
- Affects OSPF metrics for existing OSPFv2 interfaces and OSPFv2 interfaces created after the execution of this command.
- Affects OSPF metrics for only OSPFv3 interfaces created after the execution of this command.
- This command's value overrides the cost resulting from the command.
- If you want this command to apply to OSPF interfaces already configured, you need to bounce the existing interfaces: Use the **no network** and then the **network** command for the selected OSPF interfaces.
- Example 1—OSPFv2

  host1(config-router)#**ospf auto-cost reference-bandwidth 1000**

- Example 2—OSPFv3

  host1((config-router)#)#**auto-cost reference-bandwidth 1000**

- When you issue this command, the metric is calculated as follows:

  OSPF metric = bandwidth*1,000,000/link speed

  For the previous example, a 64K link yields a metric of 15625, whereas a T1 link yields a metric of 647. The minimum value for the metric is 1.

- If you never issue the **ospf auto-cost reference-bandwidth** command, OSPF calculates the cost as 10^8/link speed.
- Use the **no** version to assign cost based only on the interface type.

baseline ip ospf
baseline ipv6 ospf

- Use to set a baseline for OSPF statistics and counters.
- The following example first displays the output of the **show ip ospf** command, which is shown before you run the **baseline ip ospf** command; then it displays the execution of the **baseline ip ospf** command; and finally, it displays the **show ip ospf** command run after you execute the **baseline ip ospf** command.
  - The output of the **show ip ospf** command run before the **baseline ip ospf** command reflects the up-to-date packet counters.
  - The output of the **show ip ospf delta** command run after you run the **baseline ip ospf** command reflects the baseline set for OSPF statistics and counters.
- Example

```
host1#show ip ospf

Routing Process OSPF 1 with Router ID 5.106.7.1
 OSPF Statistics:
      Rcvd: 217935 total, 0 checksum errors
            8987 hello, 8367 database desc, 188 link state req
            159898 link state updates, 40484 link state acks
      Sent: 265026 total, 0 pkts dropped
            8927 hello, 8341 database desc, 53 link state req
            158571 link state updates, 89134 link state acks
 Supports only single TOS(TOS0) routes
 SPF schedule delay 0 secs, Hold time between two SPFs 3 secs
 Maximum path splits 1
 Area BACKBONE(0.0.0.0)
    Area is a transit area
    SPF algorithm executed 425 times
    ABR count 0
    ASBR count 1
    LSA Count 12
    Number of interfaces in this area is 24
    Area ranges are:
 Number of active areas in this router is 1
 1  normal, 0 stub, 0 NSSA.

host1#baseline ip ospf
host1#show ip ospf delta

Routing Process OSPF 1 with Router ID 5.106.7.1
 OSPF Statistics:
      Rcvd: 0 total, 0 checksum errors
            0 hello, 0 database desc, 0 link state req
            0 link state updates, 0 link state acks
      Sent: 0 total, 0 pkts dropped
            0 hello, 0 database desc, 0 link state req
            0 link state updates, 0 link state acks
 Supports only single TOS(TOS0) routes
 SPF schedule delay 0 secs, Hold time between two SPFs 3 secs
 Maximum path splits 1
```

```
Area BACKBONE(0.0.0.0)
   Area is a transit area
   SPF algorithm executed 425 times
   ABR count 0
   ASBR count 1
   LSA Count 12
   Number of interfaces in this area is 24
   Area ranges are:
Number of active areas in this router is 1
1  normal, 0 stub, 0 NSSA.
```

- There is no **no** version.

clear ipv6 ospf counters

- Use to clear all OSPF IPv6 statistical counters for the virtual router.
- Example

  host1#**clear ipv6 ospf counters**

- There is no **no** version.

clear ipv6 ospf process

- Use to clear the OSPF IPv6 process on the virtual router.
- Example

  host1#**clear ipv6 ospf process**

- There is no **no** version.

clear ip ospf database

- Use to delete all entries from the OSPF link-state database and to reset all adjacencies.
- Example

  host1#**clear ip ospf database**

- There is no **no** version.

clear ip ospf neighbor

- Use to clear an IP OSPF neighbor by specifying the IP address.

NOTE: When OSPF is configured and running over an NBMA network, do not issue the **clear ip ospf neighbor** command simultaneously on both ends of the OSPF link. Doing so brings the OSPF link down completely. In this event, you must do one of the following on both sides of the link to bring the link back up:

- Reconfigure the OSPF neighbors on the NBMA interface with the **neighbor** command.
- Issue the **clear ip ospf database** command to clear and reset the OSPF adjacencies.
- Issue the **shutdown** command followed by the **no shutdown** command on the interface.

- Example

  host1#**clear ip ospf neighbor** *neighborAddress*

- There is no **no** version.

clear ip ospf redistribution
clear ipv6 ospf redistribution

- Use to clear and readvertise all of the routes that have been previously redistributed into OSPF.

CAUTION: Using this command purges all external LSAs and reoriginates.

- Example 1

  host1#**clear ip ospf redistribution**

- Example 2

  host1#**clear ipv6 ospf redistribution**

- There is no **no** version.

default-information originate

- Use to generate a default route into an OSPF routing domain.
- When you use this command to redistribute routes into an OSPF routing domain, the router automatically becomes an AS boundary router.
- An AS boundary router, however, does not, by default, generate a default route into the OSPF routing domain. The software must have a default route before it generates one, except when you have specified the **always** keyword.

- You can specify a metric for the route or specify that the route be OSPF external type 1 or 2.
- Example

  host1(config)#**router ospf 1**
  host1(config-router)#**default-information originate route-map 5**

- Use the **no** version to disable this feature.

disable-dynamic-redistribute

- Use to halt the dynamic redistribution of routes that are initiated by changes to a route map.
- Dynamic redistribution is enabled by default.
- Example

  host1(config-router)#**disable-dynamic-redistribute**

- Use the **no** version to reenable dynamic redistribution.

distance

- Use to configure the administrative distance for OSPF routes.
- Example

  host1(config-router)#**distance ospf external 60**

- Default settings:
  - Intra-area routes—110
  - Interarea routes—112
  - External routes—114
- Use the **no** version to restore the default values.

ip ospf shutdown
ipv6 ospf shutdown

- Use to disable OSPF on the interface.
- Example 1

  host1(config-if)#**ip ospf shutdown**

- Example 2

  host1(config-if)#**ipv6 ospf shutdown**

- Use the **no** version to enable OSPF on the interface.

log-adjacency-changes
ospf log-adjacency-changes

- Use to configure the router to send a log message when the state of an OSPF neighbor changes.
- Use the **log-adjacency-changes** command for OSPFv3 interfaces; use the **ospf log-adjacency-changes** command for OSPFv2 interfaces.
- Example 1

  host1(config-router)#**log-adjacency-changes severity 3 verbosity low**

- Example 2

  host1(config-router)#**ospf log-adjacency-changes severity 3 verbosity low**

- Use the **no** version to halt logging of neighbor changes.

maximum-paths

- Use to control the maximum number of parallel routes that OSPF can support.
- The maximum number of routes can be in the range 1–16.
- The default for OSPF is 4 paths.
- To enable equal-cost multipath (ECMP) for OSPF, you need to specify a value for maximum paths greater than 1.
- Example

  host1(config-router)#**maximum-paths 2**

- Use the **no** version to restore the default value, 4.

ospf shutdown

- Use to administratively disable OSPF on the router.
- Example

  host1(config-router)#**ospf shutdown**

- Use the **no** version to reenable OSPF on the interface.

passive-interface

- Use to disable the transmission of routing updates on the interface, meaning that OSPFv2 or OSPFv3 routing information is neither sent by nor received through the interface.
- The specified interface appears as a stub network in the OSPF domain.
- By default, OSPF is enabled on a configured OSPF interface.
- Example

  host1(config-router)#**passive-interface ethernet 1/0**

- Use the **no** version to reenable the transmission of OSPF routing updates on the specified interface.

redistribute

- Use to redistribute information from a routing domain other than OSPF into the OSPF domain.
- You can set the OSPF metric type—type 1 or type 2—and set a metric for all redistributed routes.
- If you do not specify **route-map**, all routes are redistributed. By default, all routes are imported as external type 2 routes.
- If you specify **route-map** but do not list any route map tags, no routes are imported.
- Use to redistribute routes from OSPF into other non-OSPF routing domains.
- Example 1

  host1(config)#**router ospf 5**
  host1(config-router)#**redistribute bgp route-map 4**

- Example 2

  host1(config)#**router bgp 100**
  host1(config-router)#**redistribute ospf 5**

- Use the **no** version to disable redistribution.

table-map

- Use to apply a policy to modify distance, metric, metric type, route type, or tag values of OSPF routes about to be added to the IP routing table.
- The new route map is applied to all routes currently in and those subsequently placed in the forwarding table. Previously redistributed routes are redistributed with the changes caused by the route map.
- To remove from the forwarding table any old routes that are now disallowed by the specified route map, you must refresh the IP routing table with the **clear ip routes *** command.
- Example

  host1(config)#**route-map dist1 permit 5**
  host1(config-route-map)#**match community boston42**
  host1(config-route-map)#**set distance 33**
  host1(config-route-map)#**exit**
  host1(config)#**router ospf 100**
  host1(config-router)#**table-map dist1**
  host1(config-router)#**exit**
  host1(config)#**exit**
  host1#**clear ip routes ***

- Use the **no** version to halt application of the route map.

timers spf

- Use to configure the time between two consecutive SPF calculations.
- Set the time (in seconds) in the range 1–5; the default value is 3 seconds.
- If you set the hold time to 0, there is no delay between two consecutive SPF calculations. They can be done one immediately after the other.
- Example

  host1(config-router)#**timers spf 2**

- Use the **no** version to return to the default value, 3 seconds.

Default Metrics

Although the router does not support a **default-metric** command, the **redistribute** command provides two ways to set a default metric for redistributed routes.

You can simply configure a metric with the **redistribute** command to apply to all routes redistributed from the specified source protocol:

```
host1(config)#router ospf 5
host1(config-router)#redistribute bgp metric 5
```

Alternatively, you can create one or more route maps that set the metric and apply them selectively to redistributed routes:

```
host1(config)#access-list 1 permit any any
host1(config)#route-map defmetric
host1(config-route-map)#match ip address 1
host1(config-route-map)#set metric 10
host1(config-route-map)#exit
host1(config)#router ospf 5
host1(config-router)#redistribute bgp route-map defmetric
host1(config-router)#redistribute isis route-map defmetric
```

See *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*, for information about configuring route maps.

Configuring OSPF for NBMA Networks

You can configure OSPF nonbroadcast multiaccess (NBMA) networks. You can configure your OSPF network type as NBMA, regardless of the default medium. This configuration is useful when, for example, you have routers in your network that do not support multicast addressing.

You must use the **neighbor** command to specify the router's OSPF neighbors.

To configure OSPF for an NBMA network:

1. Configure an interface network type as NBMA for OSPF.

   host1(config-subif)#**ip ospf network non-broadcast**

2. Exit Interface Configuration mode. Enter Global Configuration mode.

   host1(config-subif)#**exit**

3. Configure an OSPF routing process, and enter Router Configuration mode.

   host1(config)#**router ospf 5**

4. Specify an OSPF neighbor, and optionally assign a priority number or poll interval to the neighbor.

   host1(config-router)#**neighbor 10.12.14.1 priority 5 poll-interval 180**

5. Repeat Step 4 for each neighbor in the OSPF network.

If you want to configure the network type for a specific interface or OSPF area, rather than for all OSPF interfaces, you can use the **address network** command rather than the **ip ospf network** command.

address network

- Use to configure the network type on a specific OSPF interface or for a specific OSPF area to a type other than the default for the medium.
- You must first issue the **address area** command before issuing the **address network** command.
- Example

  host1(config-router)#**address 10.12.10.2 network broadcast**

- Use the **no** version to restore the default value for the medium.

ip ospf network

- Use to configure the network type on all OSPF interfaces on the OSPF network to a type other than the default for the medium.
- Example

  host1(config-if)#**ip ospf network broadcast**

- Use the **no** version to restore the default value for the medium.

neighbor

- Use to configure OSPF neighbors on the NBMA network.
- Specify priority and poll interval only for routers that are eligible to become the designated router or backup designated router—that is, a router with a nonzero router priority value. The default priority value is 0, and the default polling interval is 120 seconds.
- Example

  host1(config-router)#**neighbor 10.12.11.5 priority 100**

- Use the **no** version to remove the neighbor or restore the default values 0 and 120.

Traffic Engineering

Traffic engineering enables more effective use of network resources by providing for the setup of explicitly routed Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) that satisfy resource and administrative constraints. You can use OSPF to exchange link resource and traffic-engineering administrative information between routers. OSPF uses this information to calculate paths in the network that satisfy the administrative constraints. MPLS can then set up LSPs along these paths. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS* for a detailed discussion of MPLS.

Configuring OSPF for Traffic Engineering

For OSPF to support traffic engineering, you must issue both of the following commands:

- **mpls traffic-eng area**—Enables the router to flood traffic engineering resource and administrative information in the specified area using type 10 opaque LSAs. These LSAs have an area-wide scope and therefore are flooded only within the indicated area.
- **mpls traffic-eng router-id**—Designates a router as traffic engineering capable and specifies the address of a stable router interface as the router ID of the node for traffic engineering purposes. The traffic engineering router ID serves as the tunnel endpoint for tunnels terminating at the node. Each node advertises its traffic engineering router ID in type 10 LSAs.

By default, OSPF always uses the MPLS tunnel to reach the MPLS endpoint. Best paths determined by SPF calculations are not considered. You can enable the consideration of best paths by issuing the **mpls spf-use-any-best-path** command. As a result, OSPF considers metrics for IGP paths and the tunnel metric, and might forward traffic along a best path, through the MPLS tunnel, or both.

You can use the **show ip ospf database opaque-area** command to display information about traffic engineering opaque LSAs.

For OSPF routes to use established MPLS tunnels as next hops—so that traffic can be mapped to use these tunnels—you must configure the tunnels with the **tunnel mpls autoroute announce ospf** command. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS*, for information about configuring MPLS on a router.

mpls spf-use-any-best-path

- Use to enable SPF calculations to consider the IGP (OSPF) best paths as well as the MPLS tunnel for forwarding traffic to the MPLS endpoint.
- By default, the MPLS tunnel is always selected for traffic to the tunnel endpoint; IGP paths are not considered. For traffic beyond the endpoint, the tunnel is considered equally with any other path.
- Example

  host1(config-router)#**mpls spf-use-any-best-path**

- Use the **no** version to disable the use of IGP best paths.

mpls traffic-eng area

- Use to enable flooding of MPLS traffic engineering link information into the specified OSPF area. Flooding is disabled by default.
- Example

  host1(config-router)#**mpls traffic-eng area 0**

- Use the **no** version to disable flooding.

mpls traffic-eng router-id

- Use to specify a stable interface to be used as a router ID for MPLS traffic engineering. Typically you specify a loopback interface to provide the greatest stability, because this is flooded to all nodes. The interface acts as the destination node for tunnels originating at other nodes.
- Example

  host1(config-router)#**mpls traffic-eng router-id loopback 0**

- Use the **no** version to remove the interface as a router ID.

Using OSPF Routes for Multicast RPF Checks

You can use the **ip route-type** or **ipv6 route-type** command to specify whether OSPF routes are available for only unicast forwarding protocols or only multicast reverse-path-forwarding (RPF) checks. Routes available for unicast forwarding appear in the unicast view of the routing table, whereas routes available for multicast RPF checks appear in the multicast view of the routing table.

To enable a multicast protocol and MPLS traffic engineering (TE) to interoperate on a router running OSPF, use the **mpls traffic-eng multicast-intact** command.

ip route-type
ipv6 route-type

- Use to specify whether OSPF routes are available only for unicast forwarding, only for multicast RPF checks, or for both.
- Use the **show ip route** or **show ipv6 route** command to view the routes available for unicast forwarding.
- Use the **show ip rpf-routes** or **show ipv6 rpf-routes** command to view the routes available for multicast RPF checks.
- By default, OSPF routes are available for both unicast forwarding and multicast RPF checks.
- Example 1

  host1(config)#**router ospf**
  host1(config-router)#**ip route-type unicast**

- Example 2

  host1(config)#**router ospf**
  host1((config-router)#)#**ipv6 route-type unicast**

- Use the **no** version to restore the default value, both.

mpls traffic-eng multicast-intact

- Use to enable a multicast protocol and MPLS traffic engineering (TE) to interoperate on a router running OSPF.
- Example

  host1(config-router)#**mpls traffic-eng multicast-intact**

- Use the **no** version to disable interoperability between a multicast protocol and MPLS-TE when running on an OSPF router.

OSPF and BGP/MPLS VPNs

Some network topologies use OSPF as the routing protocol between CE and PE routers in BGP/MPLS VPNs. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 3, Configuring BGP-MPLS Applications*, for information about configuring OSPF for this purpose.

Remote Neighbors

You can create OSPF remote neighbors to enable the router to establish neighbor adjacencies through unidirectional interfaces, such as MPLS tunnels, rather than the standard practice of using the same interface for receipt and transmission of OSPF packets. The remote neighbor can be more than one hop away through intermediate routers that are not running OSPF. OSPF uses the interface associated with the best route to reach the remote neighbor. A best route to the neighbor must exist in the IP routing table.

You must explicitly configure a remote neighbor on an OSPF router. You must specify the remote neighbor with which the router forms an adjacency and the source IP address the router uses for OSPF packets destined to its peer remote neighbor.

To form an adjacency with its remote neighbor, all OSPF packets are sent to the remote neighbor as unicast packets with the destination IP address equal to the source IP address of the remote neighbor. Use the **update-source loopback** command to assign the source IP address to a remote neighbor.

The connection between two remote neighbors is treated as an unnumbered point-to-point link that resides in the same area as that to which the pair of remote neighbors belongs.

The rules of OSPF adjacency must be followed for remote neighbors to form an adjacency with each other; for example, the neighbors must be in the same OSPF area and have the same hello interval and dead interval, and so on.

After you have used the **remote-neighbor** command to specify the remote neighbors and the update-source loopback to assign the source IP address, you must set a TTL value with the **ttl** command, because a remote neighbor can be more than one hop away. Configuration of all other remote-neighbor attributes is optional.

authentication-key

- Use to enable simple password authentication and assign a password for communication with OSPF remote neighbors.
- Example

  host1(config-router-rn)#**authentication-key 0 br549hee**

- Use the **no** version to delete the password.

authentication message-digest

- Use to specify that MD5 authentication is to be used on the OSPF remote neighbor interface.
- Example

  host1(config-router-rn)#**authentication message-digest**

- There is no **no** version.

authentication-none

- Use to specify that no authentication is to be used on the OSPF remote neighbor interface.
- Example

  host1(config-router-rn)#**authentication-none**

- There is no **no** version.

cost

- Use to specify a cost metric for the OSPF remote-neighbor interface; the metric is used in the calculation of the SPF routing table.
- The default value is 10 if there is no route to the remote neighbor; otherwise, the default is calculated based on the bandwidth of the physical interface used to reach the remote neighbor and the OSPF autocost reference bandwidth.
- Example

  host1(config-router-rn)#**cost 235**

- Use the **no** version to restore the default value.

dead-interval

- Use to set the time period, in seconds, that the OSPF router waits without receiving hello packets from a remote neighbor before declaring the neighbor to be down.
- Example

  host1(config-router-rn)#**dead-interval 180**

- Use the **no** version to restore the default value, 40 seconds.

hello-interval

- Use to set the time interval between hello packets that the router sends on the OSPF remote-neighbor interface.
- Specify a value in the range 1–65535 seconds; the default value is 40 seconds.
- Example

  host1(config-router-rn)#**hello-interval 15**

- Use the **no** version to restore the default value, 40 seconds.

message-digest-key md5

- Use to enable MD5 authentication for the OSPF remote-neighbor interface and configure the MD5 key.
- If you delete all MD5 keys, MD5 authentication is still enabled; you must either configure an MD5 key or disable MD5 authentication with the **authentication-none** command.
- Example

  host1(config-router-rn)#**message-digest-key 42 md5 0 sal29ute**

- Use the **no** version to delete the MD5 key.

remote-neighbor

- Use to configure an OSPF remote neighbor.
- Use the **update-source** command to configure source IP address for packets sent to the remote neighbor. We recommend that you do not leave the update source unconfigured for a remote neighbor.
- Example

  host1(config-router)#**remote-neighbor 10.25.100.14 area 35672**

- Use the **no** version to remove the remote neighbor and any attributes configured for the remote neighbor.

retransmit-interval

- Use to set the time between LSA retransmissions for the OSPF remote-neighbor interface when an acknowledgment for the LSA is not received.
- Specify a value in the range 1–3600 seconds; the default value is 5 seconds.
- Example

  host1(config-router-rn)#**retransmit-interval 10**

- Use the **no** version to restore the default value, 5 seconds.

transmit-delay

- Use to set the estimated time it takes to transmit a link-state update packet on the OSPF remote-neighbor interface.
- Specify a value in the range 0–3600 seconds; the default value is 1 second.
- Example

  host1(config-router-rn)#**transmit-delay 3**

- Use the **no** version to restore the default value, 1 second.

ttl

- Use to configure a hop count by setting the value of the time-to-live field used by packets sent to an OSPF remote neighbor.
- Specify a value in the range 1–255 seconds; the default value is 1 second.
- Example

  host1(config-router-rn)#**ttl 35**

- Use the **no** version to restore the default value, 1 second.

update-source

- Use to specify the loopback interface whose local IP address is used as the source address for the OSPF connection to a remote neighbor.
- We recommend that you do not leave the update source unconfigured for a remote neighbor.
- Example

  host1(config-router-rn)#**update-source loopback 1**

- Use the **no** version to delete the source address from the connection to the remote neighbor.

Remote Neighbors and Sham Links

You can configure OSPF remote neighbors to act as sham links for BGP/MPLS VPNs. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 3, Configuring BGP-MPLS Applications*, for more information.

Configuring OSPF Graceful Restart

E-series routers support OSPF graceful restart extensions as defined in RFC 3623 (Graceful OSPF Restart). Graceful restart enables a router to continue forwarding OSPF traffic based on routing information it receives prior to an unplanned restart, while the E-series router switches from the primary SRP to the secondary SRP module.

Graceful restart helps to avoid interruptions in traffic forwarding and network-wide route changes when a route processor restarts or switches over to a redundant route processor.

To accomplish OSPF graceful restart, communication must take place between the router that is restarting and its OSPF neighbors. These neighboring routers must cooperate with (or help) the restarting router by keeping it in the forwarding path while it is restarting.

The restarting router sends a grace LSA (a link-local LSA) to inform its neighbors that it is restarting. After receiving this grace LSA, the neighbors act as if the router still exists in the network topology and continue forwarding traffic through the restarting router (for the specified grace period as defined in the grace LSA). If the restarting router does not become fully adjacent with the helper router before the grace period expires, the helper abandons the helper role and determines its adjacency with the restarting router to be down. Also, based on your configuration, the helper can abandon a restart if it detects a topology change before the restart is complete.

After the router restarts, the restarting router purges the grace LSA from the OSPF domain.

To configure the router as a graceful restart helper, use the graceful restart helper commands. These commands include **graceful-restart helper** and **graceful-restart helper-abort-topology-change**.

To configure the router for a restart scenario, use the graceful restart commands. These commands include **graceful-restart**, **graceful-restart notify-time**, and **graceful-restart restart-time**.

NOTE: Graceful restart mode is supported only for OSPFv2 routers. OSPF graceful restart helper mode is supported for both OSPFv2 and OSPFv3 routers.

graceful-restart

- Use to enable OSPF graceful restart on the OSPFv2 router.
- Example

  host1(config-router)#**graceful-restart**

- Use the **no** version to disable OSPF graceful restart capability on the router.

graceful-restart helper

- Use to configure the OSPFv2 or OSPFv3 router to function as an OSPF graceful restart helper router.
- Example

  host1(config-router)#**graceful-restart helper**

- Use the **no** version to disable OSPF graceful restart helper mode capability on the router.

graceful-restart helper-abort-topology-change

- Use to specify conditions under which the OSPFv2 or OSPFv3 router abandons its role as an OSPF graceful restart helper router.
- Use the **any** keyword to abandon the helper role when any LSA changes during the restart. Use the **non-externals** keyword to abandon the helper role only when any nonexternal LSA changes during the restart.
- Example

  host1(config-router)#**graceful-restart helper-abort-topology-change any**

- Use the **no** version to return the router to its default behavior of helping a restarting OSPF router during topology changes.

graceful-restart notify-time

- Use to specify the time (in the range 1–3600 seconds) expected for the router to remove grace LSAs over all interfaces.
- The restarting router sends the sum of the restart duration and notify duration as the *grace period* to the helping neighbors in the grace LSA. Receiving a maximum-aged grace LSA is an indication to the helper that the restart has been successfully completed on the restarting router.
- If the grace period on the helper router expires before the receipt of max-aged grace LSAs, the helper router stops the restart process and does not respond to the restarting router. The helper router then originates its own LSAs with the real current state of the adjacency with the restarting router reflected in them.
- Example

  host1(config-router)#**graceful-restart notify-time 500**

- Use the **no** version to return the notify duration to its default value, 15 seconds.

graceful-restart restart-time

- Use to specify the time (in the range 1–3600 seconds) expected for the router to reacquire OSPF neighbors that were fully operational prior to the restart.
- When this timer expires, the restarting router exits the restart procedure, originates any LSAs that were suppressed during the restart, removes any self-originated LSAs that it received from helping neighbors, runs SPF, and updates any routes in the routing table.
- Example

  host1(config-router)#**graceful-restart restart-time 350**

- Use the **no** version to return the restart duration to its default value, 180 seconds.

Disabling and Reenabling Incremental SPF

By default, when changes occur to a type 5 or type 7 LSA, OSPF recalculates new, loop-free routes for only the LSAs that change. When a subset of LSAs in the external link-state database change, a full recalculation is not necessary. However, through the CLI, you can disable incremental SPF so the router can perform a full SPF on all external LSAs in the link-state database.

disable-incremental-external-spf

- Use to disable incremental external SPF on the router. When issued, this command results in a full SPF when an event occurs to trigger an external SPF.
- Example

  host1(config-router)#**disable-incremental-external-spf**

- Use the **no** version to reenable incremental SPF on this router.

Configuring OSPF Traps

You can use the **traps** command to specify OSPF traps. This command enables you to specify all or any number of the following trap settings:

- virtIfStateChange—To indicate any state change on an OSPF virtual interface
- nbrStateChange—To indicate any state change on a nonvirtual OSPF neighbor
- virtNbrStateChange—To indicate any state change on a virtual OSPF neighbor
- ifConfigErro—To indicate any configuration mismatch with a nonvirtual neighbor
- virtIfConfigError—To indicate any configuration mismatch with a virtual neighbor
- ifAuthFailure—To indicate any authentication failure on a nonvirtual interface
- virtIfAuthFailure—To indicate any authentication failure on a virtual interface
- ifRxBadPkt—To indicate the receipt of a packet that the router cannot parse
- virtIfRxBadPkt—To indicate the receipt of a packet on a virtual interface that the router cannot parse
- txRetransmit—To indicate the retransmittal of a packet on a nonvirtual interface
- virtTxRetransmit—To indicate the retransmittal of a packet on a virtual interface

- originateLsa—To indicate the origination of a new LSA by this router
- maxAgeLsa—To indicate that an LSA in this router LSDB has reached its maximum age value
- ifStateChange—To indicate a state change on an OSPF interface

traps

- Use to specify traps for OSPF.
- Example

  host1(config-router-rn)#**traps all**

- Use the **no** version to delete the specified trap, group of traps, or all traps.

Neighbor Uptime Tracking

You can use the **history** keyword with the **show ip ospf neighbors** command to display a history of up to 10 events for all OSPF neighbors or a specific OSPF neighbor. This history can aid in troubleshooting network problems related to neighbor flapping. The history includes the interface for the neighbor, a timestamp for the event, whether the neighbor transition is seen (up) or down, and the cause of down events.

You can track up to 50 neighbors; when that number is exceeded, the history of the least recently tracked neighbor is overwritten. Similarly, when a neighbor's events exceed 10, the oldest event is overwritten, because no more than 10 events can be tracked per neighbor. Neighbor uptime tracking is not available for OSPFv3. See **show ip ospf neighbors** on page 304 for output field definitions.

```
host1#show ip ospf neighbors history
Transition log for neighbor 10.10.8.2:
Interface           Event Cause           Time
============================================
ATM2/0.8            Seen  NA              WED DEC 14 07:02:27

Transition log for neighbor 10.10.12.2:
Interface           Event Cause           Time
============================================
ATM2/0.12           Seen  NA              WED DEC 14 07:09:12
ATM2/0.12           DOWN  Interface down  WED DEC 14 07:05:47
ATM2/0.12           Seen  NA              WED DEC 14 07:02:32
```

Monitoring OSPF

Two sets of commands enable you to monitor OSPF operation on your router: the **debug** and the **show** commands. Both sets of commands provide information about your router's OSPF state and configuration.

The task you are performing with each of these monitoring commands is basically the same for each command; that is, you are requesting information. The results of this request can vary. For instance, the **debug** commands provide information (some of which is dynamically obtained from router logs) about problems with the network or the router, whereas the **show** commands provide information about the actual state and configuration of your router.

debug Commands

The **debug** commands provide information about the following OSPF items:

- Adjacencies
- Designated router
- General events
- Link-state advertisements
- Neighbors
- Packets received
- Packets sent
- Route events
- SPF events

debug ip ospf
debug ipv6 ospf

- Use to display information about selected OSPF events. This command has many keywords so you can specify a variety of OSPF events.
- You can set the level of severity for the events you want displayed: 0–7.
- You can set the verbosity of the messages you want displayed: low, medium, high.
- Example 1

  host1#**debug ip ospf adj**

- Example 2

  host1#**debug ipv6 ospf lsa**

- Use the **no** version to cancel the display of any information about the designated variable.

ospf log-adjacency-changes

- Use to enable the logging of changes in the state of an OSPF neighbor.
- Example

  host1(config-router)#**ospf log-adjacency-changes**

- Use the **no** version to disable the logging of changes in the state of an OSPF neighbor.

undebug ip ospf
undebug ipv6 ospf

- Use to cancel the display of information about a selected event.
- The same OSPF variables can be designated as in the **debug ip ospf** or **debug ipv6 ospf** commands.
- Example 1

  host1#**undebug ip ospf adj**

- Example 2

  host1#**undebug ipv6 ospf lsa**

- There is no **no** version.

show Commands

The **show** commands provide information about the following OSPFv2 and OSPFv3 items:

- Routing processes
- Border routers
- Database
- Interfaces
- Neighbors
- Traffic
- Virtual links
- Internal statistics
- MPLS tunnels and opaque LSAs

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

show ip ospf
show ipv6 ospf

- Use to display general information about OSPF routing processes.
- Field descriptions
  - Routing Process—Process ID, router ID, domain ID
  - OSPF administrative state—Enabled or disabled
  - OSPF operational state—Enabled or disabled
  - Incremental External SPF—On or off
  - Graceful Restart Capability—On or off
  - Time limit to complete graceful restart—Amount of time (in seconds) during which the router can reacquire OSPF neighbors that were fully operational prior to the restart
  - Time limit to flush grace LSAs—Amount of time (in seconds) during which the router can remove grace LSAs over all interfaces
  - Graceful Restart Helper Capability—On or off
  - Graceful Restart Help:
    - Not Aborted On Topology Change
    - Aborted On Any Topology Change
    - Aborted On Any Non-External Topology Change
  - OSPF set trap field—Enabled or disabled
  - Router—Router types: internal, area border, or autonomous system boundary routers
  - OSPF Statistics—Packets received and sent; LSA discard count
  - TOS type—Number of types of service supported
  - SPF timers—Timers configured on the router
  - Maximum path splits—Maximum equal-cost paths supported
  - Areas—Areas configured and their parameters
  - Number of areas—Number of areas in the router
- Example 1

```
host1#show ip ospf

Routing Process OSPF 1 with Router ID, 0.0.0.0, Domain ID 0.0.0.0
OSPF administrative state is enabled
OSPF operational state is disabled
Incremental External SPF is ON
Graceful Restart Capability is ON
Time limit to complete graceful restart 180 seconds
Time limit to flush grace LSAs 15 seconds
Graceful Restart Helper Capability is OFF
Graceful Restart Help Not Aborted On Topology Change
Ospf set trap field disabled
OSPF Statistics:
Rcvd: 0 total, 0 checksum errors
0 hello, 0 database desc, 0 link state req
0 link state updates, 0 link state acks
```

```
Sent: 0 total, 0 pkts dropped
0 hello, 0 database desc, 0 link state req
0 link state updates, 0 link state acks
LSA discard count: 0
Supports only single TOS(TOS0) routes
SPF schedule delay 0 secs, Hold time between two SPFs 3 secs
Maximum path splits 4
Number of active areas in this router is 0
0 normal, 0 stub, 0 NSSA.
```

- Example 2

```
host1#show ip ospf

Routing Process OSPF 4 with Router ID, 10.0.0.1, Domain ID 0.0.0
 OSPF administrative state is enabled
 OSPF operational state is enabled
 Incremental External SPF is ON
 Graceful Restart Capability is OFF
 Graceful Restart Helper Capability is OFF
 Graceful Restart Help Not Aborted On Topology Change
 Ospf set trap field disabled
 OSPF Statistics:
      Rcvd: 0 total, 0 pkts dropped, 0 checksum errors
            0 hello, 0 database desc, 0 link state req
            0 link state updates, 0 link state acks
      Sent: 1 total, 0 pkts dropped
            1 hello, 0 database desc, 0 link state req
            0 link state updates, 0 link state acks
      LSA discard count: 0
 Supports only single TOS(TOS0) routes
 SPF schedule delay 0 secs, Hold time between two SPFs 3 secs
 Maximum path splits 4
 Area BACKBONE(0.0.0.0)
    SPF algorithm executed 5 times
    ABDR count 0
    ASBDR count 0
    LSA Count 1
    Number of interfaces in this area is 1
    Area ranges are:
 Area 0.0.0.1
    Area is a stub area
    Type-3 summary is filtered
    SPF algorithm executed 5 times
    ABDR count 0
    ASBDR count 0
    LSA Count 0
    Number of interfaces in this area is 0
    Area ranges are:
 Area 0.0.0.2
    Area is nssa
    Type-3 summary is filtered
    SPF algorithm executed 4 times
    ABDR count 0
    ASBDR count 0
    LSA Count 0
    Number of interfaces in this area is 0
    Area ranges are:
 Area 0.0.0.5
    Area is nssa
    SPF algorithm executed 3 times
    ABDR count 0
    ASBDR count 0
```

```
   LSA Count 0
   Number of interfaces in this area is 0
   Area ranges are:
 Number of active areas in this router is 4
 1  normal, 1 stub, 2 NSSA.
```

- Example 3

```
host1#show ipv6 ospf
Routing Process OSPFv3 1 with Router ID 10.1.1.1
 OSPFv3 administrative state is enabled
 OSPFv3 operational state is enabled
 Incremental External SPF is OFF
 Graceful Restart capability is OFF
 Graceful Restart helper capability is OFF
Ospf set trap field disabled
 SPF schedule delay 0 secs, Hold time between two SPFs 3 secs
 Maximum path splits 4
 Area BACKBONE(0.0.0.0)
SPF algorithm executed 13 times
ABDR count 1
ASBDR count 1
LSA Count 117
Number of interfaces in this area is 3
Area ranges are:
 Number of active areas in this router is 1
1   normal, 0 stub, 0 NSSA.
```

show ip ospf border-routers
show ipv6 ospf border-routers

- Use to display a list of OSPF border routers.
- Field descriptions
  - Destination—Destination's router ID
  - NEXT HOP—Next hop toward the destination
  - Interface—Interface for which you are obtaining the information
  - Router Type—Router type of the destination: either an ABR or AS boundary router, or both
  - Route Type—Type of this route: either an intra-area or interarea route
  - Area—Area ID of the area from which this route is learned
- Example 1

```
host1#show ip ospf border-routers
Destination NEXT HOP   Interface      Router Type  Route Type Area
5.5.0.250   5.5.6.250  fastethernet0  ABR/ASBR     INTRA      0.0.0.0
5.5.0 250   4.4.4.250  fastethernet0  ABR/ASBR     INTRA      0.0.0.1
6.6.6.250   4.4.4.13   fastethernet0  ABR          INTRA      0.0.0.1
```

- Example 2

```
host1#show ipv6 ospf border-routers
OSPF Area Border Routers
    Destination       NEXT_HOP       Interface  RouteType       Area
      10.0.0.10       FE80::3       ATM4/1.39   INTRA        0.0.0.0
      10.0.0.11       FE80::4       ATM4/0.41   INTRA        0.0.0.0
      10.0.0.11     FE80::5      ATM4/1.48   INTRA      100.0.0.1
```

```
OSPF Autonomous System Border Routers

    Destination       NEXT_HOP       Interface  RouteType       Area
       10.1.1.4        FE80::3       ATM4/1.39   INTER        0.0.0.0
       10.1.1.5        FE80::4       ATM4/0.41   INTER        0.0.0.0
```

show ip ospf database
show ipv6 ospf database

- Use to display the full IP OSPF database, a summary of the database, or LSAs specific to the given area.
- Field descriptions
  - Link ID—Link-state ID of the LSA; for OSPFv2:
    - For router links, set to the router's OSPF router ID
    - For network links, set to the IP interface address of the network's designated router
    - For type 3 summary LSAs, set to an IP network number
    - For type 4 summary LSAs, set to an AS boundary router ID
    - For type 5 externals, set to an IP network number
  - Link ID—Link-state ID of the LSA; for OSPFv3:
    - For link LSAs, set to the interface ID
    - For network links, set to the interface ID
    - For router links, set to integer
    - For intra-area prefix links, set to integer
    - For interarea prefix links, set to integer
    - For interarea router links, set to integer
    - For external links, set to integer
    - For grace links, set to integer
  - ADV Router—ID of the advertising router
  - Age—Link-state age
  - Seq#—Link-state sequence number (detects old or duplicate LSAs)
  - Checksum—Fletcher checksum of the complete contents of the LSA
  - Area—Area for which data is displayed
  - Router—Number of router LSAs
  - Network—Number of network LSAs
  - Intra-Prefix—Number of intra-prefix LSAs
  - Inter-Prefix—Number of inter-prefix LSAs
  - Inter-Router—Number of inter-outer LSAs
  - Link LSAs—Number of link LSAs
  - Grace LSAs—Number of graceful restart LSAs
  - External LSAs—Number of external LSAs

- MaxAge—Number of LSAs that have reached the maximum age
- Area—Area for this LSA
- LS age—LSA age
- Options—Optional capabilities supported by this router
- LS Type—LSA type
- Link State ID—Link-state ID of the link local LSA
- Length—Length of the LSA (in bytes)
- Bit set—Bit set used by this LSA type
- Link connected to—Type of network to which the link connects
- Neighboring router's Router ID—Router ID of the neighboring router
- Neighboring router's Interface ID—Interface ID of the neighboring router
- Local Interface ID—Local interface ID
- Metric—Cost of this interface
- Attached Router—Addresses of any attached routers
- Router Priority—Priority value configured for the router
- Link Local Address—Originating router's link-local interface address on the link
- Prefixes—Prefixes associated with this LSA
- Number of Prefixes—Number of prefixes associated with this LSA
- Referenced LSA Type—Router LSA or network LSA with which the IPv6 address prefixes should be associated
- Referenced LSA Advertising Router—Router LSA or network LSA with which the IPv6 address prefixes should be associated
- Referenced LSA ID—Router LSA or network LSA with which the IPv6 address prefixes should be associated
- asbr—Address of the AS boundary router
- LS Seq Number—Sequence number of the LSA
- TLVs—Type of TLV included in LSA
  - 1(Restart duration)—Duration of the restart, in seconds
  - 2(Restart Reason)—Reason that the peer restarted: Unknown, Software Restart, Software Reload, Software Upgrade, Switch to redundant control processor
  - 3(Unknown)—Any recognized type is listed as type 3, unknown; consequently the meaning and units of the value are unknown as well
- length—Length of the TLV; varies according to the TLV
- Value—Value of the TLV; varies according to TLV

- Example 1—OSPFv2 output

```
host1#show ip ospf database
OSPF Database

  Router Link States (Area 0.0.0.0)

  Link ID    ADV Router   Age  Seq#        Checksum
5.1.101.1    5.1.101.1    932  0x80000069  0x102f
192.168.1.13 192.168.1.13 1763 0x80000099  0xaa4e
192.168.1.10 192.168.1.10 285  0x80000087  0xada6
192.168.1.11 192.168.1.11 401  0x80000087  0xaba5
192.168.24.6 192.168.24.6 622  0x800005bf  0x6087

Network Link States (Area 0.0.0.0)

  Link ID    ADV Router    Age Seq#       Checksum
56.56.56.220 5.6.6.1       499 0x80000069 0x26a0
192.168.1.12 192.168.254.6 622 0x8000009e 0xebc2

  Summary Link States (Area 0.0.0.0)

  Link ID ADV Router   Age Seq#       Checksum
4.4.4.0   5.5.0.250    497 0x8000005a 0x2ca6
4.4.4.0   192.168.1.13 528 0x80000059 0x 45d

  AS Summary Link States (Area 0.0.0.0)

  Link ID ADV Router   Age Seq#       Checksum
5.5.0.250 192.168.1.13 491 0x80000002 0xe9d4

  AS External Link States

Link ID ADV Router Age Seq#       Checksum
8.8.8.0 5.5.0.250  502 0x8000005f 0x2d67

  Router Link States (Area 0.0.0.1)

Link ID      ADV Router   Age Seq#       Checksum
5.5.0.250    5.5.0.250    498 0x80000067 0xdec1
192.168.1.13 192.168.1.13 505 0x800000a5 0x3b32

Network Link States (Area 0.0.0.1)

Link ID     ADV Router     Age Seq#        Checksum
4.4.4.13    192.168.1.13   505 0x80000001  0x410b
5.1.0.0     192.168.1.13   940 0x80000059  0x82c4
5.2.0.0     5.5.0.250      495 0x80000001  0x51bf
5.2.0.0     192.168.1.13   932 0x80000059  0x76cf
5.3.0.0     5.5.0.250      495 0x80000001  0x45ca
5.3.0.0     192.168.1.13   932 0x80000059  0x6ada
56.56.56.0  5.5.0.250      495 0x80000062  0xc469

AS Summary Link States (Area 0.0.0.1)

Link ID   ADV Router Age Seq#         Checksum
5.5.0.250 5.5.0.250  496 0x800000001  0x51c0
```

- Example 2—OSPFv3 general output

```
host1#show ipv6 ospf database
OSPF Database
     V3 Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         2.2.2.2         167         0x80000003  0xa9e3
0.0.0.0         3.3.3.3         168         0x80000002  0x2c63

      V3 Inter-Area Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.1         2.2.2.2         33          0x80000004  0x5288

      V3 Inter-Area Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.1         2.2.2.2         33          0x80000001  0x a0f

      V3 Intra-Area Prefix Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.1         2.2.2.2         167         0x80000003  0xc8ba
0.0.0.1         3.3.3.3         168         0x80000003  0xdc9e

      V3 Link Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#       Checksum
50.0.0.10       2.2.2.2         178         0x80000001  0xb51d
50.0.0.13       3.3.3.3         178         0x80000001  0x8c3e

     V3 Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         1.1.1.1         40          0x80000003  0xf7a4
0.0.0.0         2.2.2.2         168         0x80000003  0x7825

      V3 Inter-Area Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.2         2.2.2.2         33          0x80000004  0x6a4f

      V3 Intra-Area Prefix Link States (Area 0.0.0.1)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.1         1.1.1.1         169         0x80000003  0x911a
0.0.0.1         2.2.2.2         168         0x80000003  0xa5fd

      V3 Link Link States (Area 0.0.0.1)

Link ID         ADV Router      Age         Seq#       Checksum
50.0.0.6        1.1.1.1         180         0x80000001  0x44b7
50.0.0.9        2.2.2.2         178         0x80000001  0x1bd8

      V3 External Link States

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.1         1.1.1.1         40          0x80000001  0xe5a0
```

- Example 3—OSPFv3 database summary information

```
host1:v2#show ipv6 ospf database database-summary
 Area     Router   Network   Intra-Prefix   Inter-Prefix   Inter-Router
-------   ------   -------   ------------   ------------   ------------
0.0.0.0   2        1         3              0              0
 Area     MaxAge
-------   ------
0.0.0.0   0

Link LSAs: 2, Max age: 0
Grace LSAs: 1, Max age: 0
External LSAs: 0, Max age: 0
```

- Example 4—OSPFv3 LSA output (router)

```
host1#show ipv6 ospf database router
V3 Router Link States (Area 0.0.0.0)
  LS age: 433
  Options: (  V6-Bit , R-Bit , ExternalRoutingCapability,  No Nssa-LSA)
  LS Type: Router Links
  Link State ID: 0.0.0.0
  Advertising Router: 1.1.1.1
  LS Seq Number: 0x80000002
  Checksum: 0x c90
  Length: 40
  Bit E set
     Link connected to: a Point To Point Network
      Neighboring  router's Router Id: 2.2.2.2
      Neighboring  router's Interface Id: 0x3200000a
      Local Interface ID : 0x32000006
      Metric 1

  LS age: 432
  Options: (  V6-Bit , R-Bit , ExternalRoutingCapability,  No Nssa-LSA)
  LS Type: Router Links
  Link State ID: 0.0.0.0
  Advertising Router: 2.2.2.2
  LS Seq Number: 0x80000002
  Checksum: 0x8519
  Length: 40

     Link connected to: a Point To Point Network
      Neighboring  router's Router Id: 1.1.1.1
      Neighboring  router's Interface Id: 0x32000006
      Local Interface ID : 0x3200000a
      Metric 1
```

- Example 5—OSPFv3 LSA output (network)

```
host1#show ipv6 ospf database network
        (Area 0.0.0.1)
    LS Type: Network LSA
    Link State ID: 0.0.0.14
    Advertising Router: 3.3.3.3
      LS age: 131
    LS Seq Number: 0x80000001
    Checksum: 0x6c69
    Length: 32
Options:   V6-bit set, ExternalRoutingCapability, R-bit set
      Attached Router: 3.3.3.3
      Attached Router: 2.2.2.2
```

- Example 6—OSPFv3 LSA output (link)

```
host1#show ipv6 ospf database link
V3 Link Link States (Area 0.0.0.0)

Link ID         ADV Router      Age         Seq#        Checksum

  LS age: 280
  LS Type: Link
  Link State ID: 0x32000006
  Advertising Router: 1.1.1.1
  LS Seq Number: 0x80000001
  Checksum: 0x44b7
  Length: 56
  Router Priority 0
  Link Local Address fe80::90:1a00:200:670
  Prefixes
       1:1:1:1000:: / 60       options 0       metric 0

  LS age: 282
  LS Type: Link
  Link State ID: 0x3200000a
  Advertising Router: 2.2.2.2
  LS Seq Number: 0x80000001
  Checksum: 0x11e1
  Length: 56
  Router Priority 0
  Link Local Address fe80::90:1a00:300:670
  Prefixes
       1:1:1:1000:: / 60       options 0       metric 0
```

- Example 7—OSPFv3 LSA output (intra-area-prefix)

```
host1#show ipv6 ospf database intra-area-prefix

      V3 Intra Area Prefix Link States (Area 0.0.0.0)
  LS age: 162
  LS Type: Intra Area Prefix Links
  Link State ID: 0.0.0.1
  Advertising Router: 1.1.1.1
  LS Seq Number: 0x80000003
  Checksum: 0x911a
  Length: 44
  Number of Prefixes 1
  Referenced LSA Type 0x    2001
  Referenced LSA Advertising Router 1.1.1.1
  Referenced LSA ID 0
  Prefixes
       1:1:1:1000:: / 60       options 0       metric 1

  LS age: 161
  LS Type: Intra Area Prefix Links
  Link State ID: 0.0.0.1
  Advertising Router: 2.2.2.2
  LS Seq Number: 0x80000003
  Checksum: 0xa5fd
  Length: 44
  Number of Prefixes 1
  Referenced LSA Type 0x    2001
  Referenced LSA Advertising Router 2.2.2.2
  Referenced LSA ID 0
  Prefixes
       1:1:1:1000:: / 60       options 0       metric 1
```

- Example 8—OSPFv3 LSA output (interarea router)

```
host1#show ipv6 ospf database inter-area-router

      V3 Inter-Area-Router Link States (Area 0.0.0.0)
  LS age: 304
  LS Type: Inter Area Net Links
  Link State ID: 0.0.0.1
  Advertising Router: 2.2.2.2
  LS Seq Number: 0x80000001
  Checksum: 0x a0f
  Length: 32
     Metric: 1
     Options: 19
     asbr: 1.1.1.1
```

- Example 9—OSPFv3 LSA output (graceful restart helper)

```
host1#show ipv6 ospf database grace
      V3 Grace Link States (Area 0.0.0.1)
LS age: 3
  LS Type: Grace
  Link State ID: 0x00000002
  Advertising Router: 2.2.2.2
  LS Seq Number: 0x80000001
  Checksum: 0x8409
  Length: 44
 TLVs
Type: 1(Restart duration), length: 4, Value: 150
Type: 2(Restart Reason), length: 1, Value: 2(Software Reload)
Type: 3(Unknown), length: 4, Value: 33686018
```

show ip ospf database link-local

- Use to display OSPF database link local states.
- Field descriptions
  - Interface—Interface for which you are obtaining link-local LSA
  - LS age—Age of LSA
  - LS Type—Type of LSA (Link Local)
  - Link State ID—Link-state ID of the link local LSA
  - Advertising Router—Router ID of the router that originated the LSA
  - LS Seq Number—Link-state sequence number to identify duplicate or old LSIDs
  - Checksum—Checksum of the complete contents of the LSA
  - Length—Length of the LSA in bytes
  - Opaque LSA Type—Type of opaque LSA
  - Neighbor—Neighbor IP address
  - Grace Period—Helper grace period in seconds
  - Restart Reason—Reason for restart; Planned Restart or Unplanned Restart

- Example

```
host1#show ip ospf database link-local
Link-Local States

 Interface : ATM1/3.80
  LS age: 17
  LS Type: Link Local
  Link State ID: 3.0.0.0
  Advertising Router: 100.1.1.67
  LS Seq Number: 0x80000002
  Checksum: 0xac91
  Length: 36
  Opaque LSA Type : Restart Grace
  Neighbor 0.0.0.0
  Grace Period 90 seconds
  Restart Reason : Unplanned Restart
```

show ip ospf database opaque-area

- Use to display lists of information about the TE opaque LSAs.
- The TE router address LSA describes a stable IP address on the originating router that can be used for TE purposes—such as setting up TE LSPs to this address.
- The TE link LSA describes TE information about an interface on the originating router.
- Field descriptions
  - LS age—Age of LSA
  - Options—Optional capabilities supported by the described portion of the routing domain
  - LS Type—Type of LSA; opaque area TE router address or opaque area TE link LSA
  - Link State ID—Link-state ID of the opaque LSA
  - Advertising Router—Router ID of the router that originated the LSA
  - LS Seq Number—Link-state sequence number to identify duplicate or old LSIDs
  - Checksum—Checksum of the complete contents of the LSA
  - Length—Length of the LSA in bytes
  - TE Router-ID—Traffic engineering router ID of the originating router
  - Link Type—Point-to-point or multiaccess
  - Link ID—For point-to-point interfaces, this is the router ID of the router at the remote end; for multiaccess interfaces, this is the address of the DR
  - Local Address—IP address of the local interface for the link
  - Remote Address—IP address of the remote (neighbor's) interface for the link
  - TE Metric—Link metric for traffic engineering purposes; can be different from the standard OSPF link
  - Max BW—Maximum bandwidth that can be used on this link in this direction

- Max Reservable BW—Maximum bandwidth that can be reserved on this link; can exceed the maximum bandwidth in the event of oversubscription
- Max Unreserved BW—Amount of bandwidth not yet reserved at each of the eight priority levels; each value is less than or equal to the maximum reservable bandwidth
- Color—Bitmask that specifies the administrative group membership for this link; a link that is a member of more than one group will have multiple bits set

- Example

```
host1#show ip ospf database opaque-area

      Opaque-area Link States (Area 0.0.0.0)

  LS age: 914
  Options:   (TOS-capable, No Type7-LSA, ExternalRoutingCapability, No
  Multicast Capability, No External Attributes    LSA)
  LS Type: Opaque-Area (TE Router Address)
  Link State ID: 1.0.0.0(Instance)
  Advertising Router: 100.1.1.1
  LS Seq Number: 0x80000003
  Checksum: 0xd293
  Length: 28
  TE Router-ID: 100.1.1.1

  LS age: 919
  Options:   (TOS-capable, No Type7-LSA, ExternalRoutingCapability, No
  Multicast Capability, No External Attributes    LSA)
  LS Type: Opaque-Area (TE Links)
  Link State ID: 1.0.0.1(Instance)
  Advertising Router: 100.1.1.1
  LS Seq Number: 0x80000003
  Checksum: 0xf66e
  Length: 124
  Link Type: P2P
  Link ID: 1744896257
 Local Address 14.1.1.2
 Remote Address 14.1.1.1
 TE Metric 0
 Max BW 1000 kb/sec (125000 Bps)
 Max Reservable BW 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 0 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 1 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 2 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 3 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 4 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 5 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 6 1000 kb/sec (125000 Bps)
 Max Unreserved BW : pri 7 1000 kb/sec (125000 Bps)
 Color 0
```

show ip ospf interface
show ipv6 ospf interface

- Use to display a list of OSPFv2 or OSPFv3 interfaces.
- Use the optional *areaId* or *areaIdInt* values to specify an OSPF area ID in either IP or decimal format.
- Field descriptions
  - Interface value (fastEthernet)—Status of the physical link and the operational status of the protocol
  - Internet Address—Interface IP address
  - Area—Area identifier: IP address
  - Network type—Broadcast, NBMA, Point-to-Point, or Point-to-Multipoint
  - Authentication type—None, simple, or MD5
  - Cost—Metric for OSPF transmission
  - Transmit Delay—Time between transmissions from the specified interface
  - Interface State—Current state of the specified interface
  - Priority—Router's priority on the specified interface
  - Designated Router—Designated router ID and respective interface IP address
  - Backup Designated Router—Designated router ID and respective interface IP address of the backup router
  - Timer intervals—Configuration of timer intervals: Hello, Dead, Wait, and Retransmit
  - Neighbor Count—Number of neighbors and their state; adjacent neighbors
  - LDP is configured through LDP autoconfig—Indicates whether LDP is configured on the interface by means of autoconfiguration; supported only for OSPFv2
  - LDP-IGP Synchronization—Status of synchronization, Achieved or Pending; supported only for OSPFv2
- Example 1

```
host1#show ip ospf interface
FastEthernet0 is up, OSPF line protocol is up
OSPF interface configuration:
  Internet Address 192.168.1.250, Area 0.0.0.0
  Network type BROADCAST, No authentication
  Cost: 1
  Transmit Delay is 1 sec, Interface State DROTHER, Priority 1
  Designated Router (Interface address) 192.168.1.107
  Backup Designated Router (Interface address) 192.168.1.214
  Timer intervals configured, Hello 10, Dead 40, Wait 120, Retransmit 5
  Neighbor Count is 2, Adjacent neighbor count is 2
  Adjacent with neighbor 192.168.1.107 (Designated Router)
  Adjacent with neighbor 192.168.254.7 (Backup Designated Router)
  LDP is configured through LDP autoconfig
  LDP-IGP Synchronization: Achieved
```

- Example 2

```
host1#show ipv6 ospf interface
ATM4/0.12 is up, OSPFv3 line protocol is up
 Area 0.0.0.0, Intf ID: 0x320004, Instance ID: 0
 Link Local Address: fe80::90:1a00:100:80
 Interface is active
 Network type POINT-TO-POINT
 Interface State POINT-TO-POINT
 Cost: 1, Priority 1
 No designated router on this network
 No backup designated router on this network
 Timer intervals configured:
 Hello 10, Dead 40, Wait 40
 Transmit Delay is 1 sec(s)
 Retransmit interval is 5 secs
 Neighbor Count is 1
  FULL Adjacent neighbor count is 1
  Adjacent with neighbor 11.0.0.2

FastEthernet0/0 is up, OSPFv3 line protocol is up
OSPF interface configuration:
 Interface ID 0.0.1.1
 IPv6 link-local address FE80::3/128
 IPv6 prefix address 3000::1/64, Area 0.0.0.0
 Network type BROADCAST
 Cost: 1
 Transmit Delay is 1 sec, Interface State BACKUPDR, Priority 1
 Designated Router's router ID 1.1.1.1
 Backup Designated Router's router ID 2.2.2.2
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
 Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 3.3.3.3 (Designated Router)
```

show ip ospf internal-statistics
show ipv6 ospf internal-statistics

- Use to display internal OSPFv2 or OSPFv3 statistics, such as allocation failures for different OSPF components.
- Use the **delta** keyword to display statistics relative to the current baseline.
- Field descriptions
  - LSA bytes allocated—Number of bytes allocated for LSAs
  - Router LSA bytes allocated—Number of bytes allocated for router LSAs
  - Summary bytes allocated—Number of bytes allocated for summary LSAs
  - Neighbor RTX bytes allocated—Number of bytes allocated for neighbor retransmissions
  - Timers bytes allocated—Number of bytes allocated for OSPF timers
  - Ospf total bytes free—Total number of bytes free
  - Ospf heap total bytes allocated—Total number of bytes allocated from the OSPF heap
  - Neighbor allocation failures—Number of neighbor allocation failures
  - LSA allocation failures—Number of LSA allocation failures
  - LSA HDR allocation failures—Number of LSA header allocation failures

- DB Request allocation failures—Number of database request allocation failures
- RTX allocation failures—Number of neighbor retransmission allocation failures
- LS Ack allocation failures—Number of LSA acknowledgment packet allocation failures
- DD pkt allocation failures—Number of database description packet allocation failures
- OSPF interface allocation failures—Number of interface allocation failures
- OSPF general packet allocation failures—Number of general packet allocation failures

- Example 1

```
host1#show ip ospf internal-statistics
 Routing Process OSPF 1 with Router ID 5.72.3.1
 Internal OSPF Statistics, bytes allocated/free:
      LSA bytes allocated:216
      Router LSA bytes allocated:936
      Summary bytes allocated:0
      Neighbor RTX bytes allocated:0
      Timers bytes allocated:352
      Ospf total bytes free:824368
      Ospf heap total bytes allocated:1048576
 Internal OSPF Statistics, allocation failures:
      Neighbor allocation failures:0
      LSA allocation failures:0
      LSA HDR allocation failures:0
      DB Request allocation failures:0
      RTX allocation failures:0
      LS Ack allocation failures:0
      DD pkt allocation failures:0
      OSPF interface allocation failures:0
      OSPF general packet allocation failures:0
```

- Example 2

```
host1#show ipv6 ospf internal-statistics
Routing Process OSPFv3 1 with Router ID 1.1.1.1
 Internal OSPF Statistics, bytes allocated/free:
      LSA bytes allocated:                    39
      Router LSA bytes allocated:             1314774
      Summary bytes allocated:                0
      Timers bytes allocated:                 96
      Ospf total bytes free:                  16
      Ospf heap total bytes allocated:        1000
 Internal OSPF Statistics, allocation failures:
      Neighbor allocation failures:           0
      LSA allocation failures:                0
      LSA HDR allocation failures:            0
      DB Request allocation failures:         0
      RTX allocation failures:                0
      LS Ack allocation failures:             0
      DD pkt allocation failures:             0
      OSPF interface allocation failures:     0
      OSPF general packet allocation failures: 0
```

show ip ospf neighbors
show ipv6 ospf neighbors

- Use to display information about OSPF neighbors on a per-interface basis.
- Use the optional *areaId* or *areaIdInt* values, in the **show ipv6 ospf neighbors command**, to specify an OSPFv3 area ID in either IP or decimal format.
- You can use the **history** keyword with the **show ip ospf neighbors** command to display a history of up to 10 events for all OSPF neighbors or a specific OSPF neighbor. This neighbor uptime tracking feature is not available for OSPFv3. For more information, see *Neighbor Uptime Tracking* on page 286.
- Field descriptions
  - Neighbor ID—Neighbor's router ID
  - Pri—Router priority of neighbor
  - State—OSPF neighbor's state
    - DR—Designated router
    - BDR—Backup designated router
    - DR Other—Neighbor to a designated router or a backup designated router
  - Dead Time—Interval since last hello packet from neighbor
  - Address—IP address of the neighbor's interface
  - Intf ID—Interface ID of the neighbor's interface
  - Interface—Name of the specified interface and its port number
  - Transition log—List of transitions events for a neighbor
  - Interface—Interface for the neighbor
  - Event—Transition event
  - Cause—Cause of transition event
  - Time—Time stamp for the event in *day month date HH*:*MM*:*SS* format
- Example 1

```
host1#show ip ospf neighbors
    Neighbor ID  Pri  State            Dead Time     Address    Interface
10.0.8.1     1    TWO-WAY/DR Other  00:00:39      10.0.76.1  fastEthernet11/0
10.0.71.1    1    FULL/DR           00:00:42      10.0.76.2  fastEthernet11/0
10.0.96.1    1    FULL/BDR          00:00:28      10.0.76.4  fastEthernet11/0
```

- Example 2

```
host1#show ipv6 ospf neighbors
    Neighbor ID  Pri  State           Dead Time Intf ID     Interface
1.1.1.1      1    TWO-WAY/DROTHER  00:00:40  0x3200042a  FastEthernet13/1.172
3.3.3.3      1    FULL/BDR         00:00:40  0x32000494  FastEthernet13/1.172
4.4.4.4      1    FULL/DR          00:00:40  0x320004c9  FastEthernet13/1.172
```

- Example 3

```
host1#show ip ospf neighbors history
Transition log for neighbor 10.10.8.2:
Interface           Event Cause           Time
=============================================
ATM2/0.8            Seen  NA              WED DEC 14 07:02:27

Transition log for neighbor 10.10.12.2:
Interface           Event Cause           Time
=============================================
ATM2/0.12           Seen  NA              WED DEC 14 07:09:12
ATM2/0.12           DOWN  Interface down  WED DEC 14 07:05:47
ATM2/0.12           Seen  NA              WED DEC 14 07:02:32
```

show ip ospf remote-neighbor interface

- Use to display all interfaces that are associated with OSPF remote neighbors.
- Field descriptions
  - OSPF remote-neighbor—Remote neighbor address for this interface
  - Update-source—Update source for this interface
  - Remote-neighbor reachable—Reachable status of the remote neighbor, yes or no
  - Area—Area of this interface
  - Network type—Network type for this interface
  - Cost—Cost value for this interface
  - Transmit Delay—Transmit delay for this interface, in seconds
  - Interface State—Interface state
  - Priority—Priority value for this interface
  - Designated router—Designated router on this network, if any
  - Backup designated router—Backup designated router on this network, if any
  - Hello—Hello timer value, in seconds
  - Dead—Dead interval timer value, in seconds
  - Wait—Wait interval timer value, in seconds
  - Retransmit—Retransmit interval timer value, in seconds
  - Neighbor Count—Number of neighbors to this interface
  - Adjacent neighbor count—Number of adjacent neighbors to this interface
  - Adjacent with neighbor—Address of the neighbor adjacent to this interface
- Example

```
host1#show ip ospf remote-neighbor interface
OSPF remote-neighbor 221.221.221.221 interface configuration:
 Update-source loopback0
 Remote-neighbor reachable: yes
 Area 0.0.0.0
 Network type POINT-TO-POINT, No authentication
 Cost: 1
```

```
Transmit Delay is 1 sec, Interface State POINT-TO-POINT, Priority 1
No designated router on this network
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Neighbor Count is 1, Adjacent neighbor count is 1
 Adjacent with neighbor 221.221.221.221
```

show ip ospf spf-log

- Use to display how often and why the router has run a full SPF calculation.
- Field descriptions
  - Intra SPF log—Log for SPF calculations run to compute intra-area LSAs
  - Inter SPF log—Log for SPF calculations run to compute interarea LSAs
  - External SPF log—Log for SPF calculations run to compute routes outside the OSPF routing domain
  - When—Amount of time since a full SPF calculation took place, in *hours:minutes:seconds*; the previous 20 calculations are logged
  - Duration—Number of milliseconds to complete this SPF run; the elapsed time is in actual clock time, not CPU time
  - LSA Router Id—Whenever a full SPF calculation is triggered by a new LSA, the router ID is stored in the router
  - Triggers—List of reasons that triggered a full SPF calculation
- Example

```
host1#show ip ospf spf-log

Intra SPF log
When     Duration      LSA Router Id    Triggers
00:04:42 0.000         23.23.23.3       Protocol Off
00:04:38 0.000         23.23.23.3       LSA Add
00:04:34 0.000         12.12.12.2       LSA Add
00:04:30 0.010         23.23.23.3       LSA Update
00:03:51 0.000         23.23.23.3       Protocol Off
00:03:47 0.000         23.23.23.3       LSA Add
00:03:43 0.000         12.12.12.2       LSA Add
00:03:39 0.000         23.23.23.3       LSA Update

Inter SPF log
When     Duration      LSA Router Id    Triggers
00:04:46 0.010         23.23.23.3       Protocol Off
00:04:42 0.000         23.23.23.3       LSA Add
00:04:38 0.000         12.12.12.2       LSA Add
00:04:34 0.000         23.23.23.3       LSA Update
00:03:55 0.000         23.23.23.3       Protocol Off
00:03:51 0.000         23.23.23.3       LSA Add
00:03:47 0.000         12.12.12.2       LSA Add
00:03:43 0.000         23.23.23.3       LSA Update
```

```
External SPF log
When     Duration      LSA Router Id    Triggers
00:04:47 0.000         23.23.23.3       Protocol Off
00:04:43 0.000         23.23.23.3       LSA Add
00:04:39 0.000         12.12.12.2       LSA Add
00:04:35 0.010         23.23.23.3       LSA Update
00:03:56 0.000         23.23.23.3       Protocol Off
00:03:52 0.000         23.23.23.3       LSA Add
00:03:48 0.000         12.12.12.2       LSA Add
00:03:44 0.000         23.23.23.3       LSA Update
```

show ipv6 ospf summary-prefix

- Use to display summary prefixes configured to summarize externals.
- Example

```
host1#show ipv6 ospf summary-prefix
Summary Prefixes
4:: / 64
5:: / 64
```

show ipv6 ospf traffic

- Use to display OSPFv3 packet statistics.
- Use the **delta** keyword to display statistics relative to the current baseline.
- Field descriptions
  - Rcvd
    - total—Total number of packets received
    - checksum errors—Total number of packets received that contained checksum errors
    - hello—Total number of hello packets received
    - database desc—Total number of database description packets received
    - link state req—Total number of link-state request packets received
    - link state updates—Total number of link-state update packets received
    - link state acks—Total number of link-state acknowledge packets received
  - Sent
    - total—Total number of sent packets
    - pkts dropped—Total number of packets dropped
    - hello—Total number of hello packets sent
    - database desc—Total number of database description packets sent
    - link state req—Total number of link-state request packets sent
    - link state updates—Total number of link-state update packets sent
    - link state acks—Total number of link-state acknowledge packets sent
  - LSA discard count—Total number of packets discarded

- Example

```
host1#show ipv6 ospf traffic
OSPFv3 Statistics:
      Rcvd: 249 total, 0 checksum errors
            242 hello, 2 database desc, 1 link state req
            4 link state updates, 1 link state acks
      Sent: 251 total, 0 pkts dropped
            242 hello, 3 database desc, 1 link state req
            4 link state updates, 1 link state acks
      LSA discard count: 0
```

show ip ospf virtual-links

- Use to display the parameters and the current state of OSPF virtual links.
- Field descriptions
  - Virtual link to router—OSPF neighbor and the current state of the virtual link
  - Transmit Delay—Time (in seconds) between transmissions from the specified interface
  - Timer intervals—Timer intervals (in seconds) configured for the link: Hello, Dead, and Retransmit
- Example

```
host1#show ip ospf virtual-links
Virtual link to router 192.168.1.13 in state POINT-TO-POINT
Transmit Delay is 1 sec
Timer intervals configured, Hello 10 sec, Dead 40 sec, Retransmit 5 sec
```

Chapter 6

Configuring IS-IS

This chapter describes how to configure Intermediate System–to–Intermediate System (IS-IS) routing on your E-series router; it contains the following sections:

Overview

IS-IS is a dynamic routing protocol developed by the International Organization for Standardization (ISO) and commonly referred to as ISO 10589. IS-IS was originally developed at Digital Equipment Corporation for Phase V DECnet. The motivation to standardize IS-IS, however, was through the efforts of the American National Standards Institute (ANSI) X3S3.3 Network and Transport Layers Committee.

Similar to the Open Shortest Path First (OSPF) routing protocol, IS-IS is a link-state protocol. It builds a complete and consistent picture of a network's topology by sharing link-state information across all network Intermediate System (IS) devices.

The IS-IS routing protocol provides routing for pure Open Systems Interconnection (OSI) environments. IS-IS as implemented on the E-series router supports IP networks and enables you to configure IS-IS as an IP routing protocol only. In IS-IS, networks are partitioned into routing domains, which are further divided into areas. A two-level hierarchical routing design is used. With this model, routing is referred to as level 1, level 2, or both level 1 and level 2.

IS-IS Terms

OSI internetworking has its own terminology. A number of terms used in IS-IS routing discussions are defined in Table 13.

Table 13: IS-IS Terms

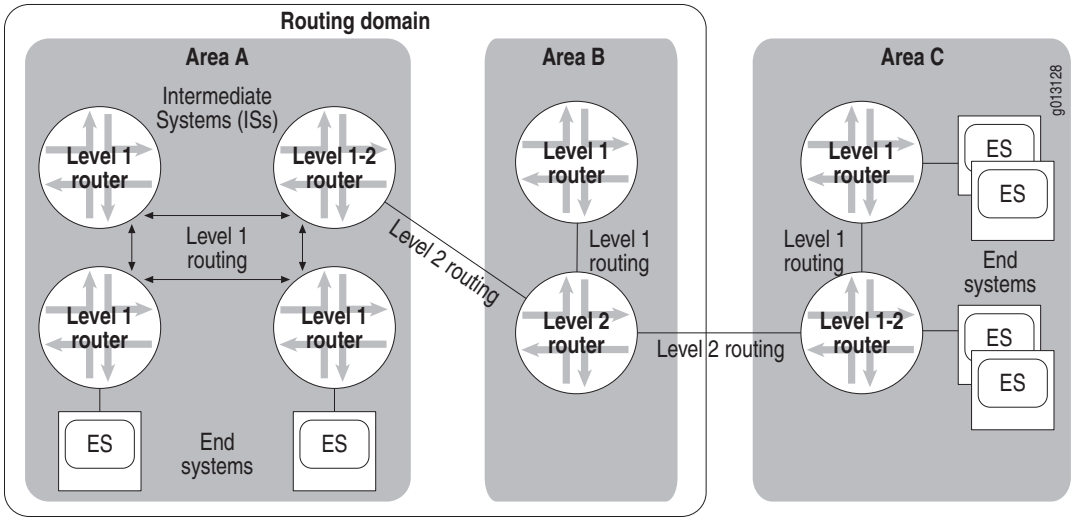
Term	Meaning
area	A group of contiguous networks and their attached hosts. Area boundaries are normally assigned by a network administrator.
complete sequence number PDU (CSNP)	PDU sent by designated router to ensure database synchronization
Connectionless Network Protocol (CLNP)	An OSI network layer protocol used by CLNS to handle data at the transport layer; the OSI equivalent of IP
Connectionless Network Service Protocol (CLNS)	An OSI network layer service that enables data transmission without establishing a circuit and that routes messages independently of any other messages.
end system (ES)	Any nonrouting network node or host
intermediate system (IS)	A router
level 1 routing	■ Routing *within* an area ■ Level 1 routers (or intermediate systems) track all the individual links, routers, and end systems within a level 1 area. ■ Level 1 routers do not know the identity of routers or destinations outside their area. ■ A level 1 router forwards all traffic for destinations outside its area to the nearest level 2 router within its area.

Table 13: IS-IS Terms (continued)

Term	Meaning
level 2 routing	■ Routing *between* areas ■ Level 2 routers know the level 2 topology and know which addresses are reachable via each level 2 router. ■ Level 2 routers track the location of each level 1 area. ■ Level 2 routers are not concerned with the topology within any level 1 area (for example, the details internal to each level 1 area). ■ Level 2 routers can identify when a level 2 router is also a level 1 router within the same area. ■ Only a level 2 router can exchange packets with external routers located outside its routing domain.
link-state PDU (LSP)	PDU broadcast by link-state protocols that contains information about neighbors and path costs; used to maintain routing tables; also known as link-state advertisement
network entity title (NET)	ISO network addresses used by CLNS networks; an identifier of a network entity in an end system or intermediate system. A NET consists of an area address (routing domain), system identifier, and selector.
network service access point (NSAP)	Hierarchical network address that specifies the point at which network services are made available to a transport layer entity in the OSI reference model. A valid NSAP address is unique and unambiguously identifies a single system.
partial sequence number PDU (PSNP)	PDU sent by designated router to acknowledge and request link-state information
protocol data unit (PDU)	OSI term equivalent to packet, containing protocol control information and, possibly, user data. This chapter uses the term packet interchangeably with PDU.
route tag	A numeric value assigned to the IP addresses on an IS-IS route before the route is propagated to other routers in an IS-IS domain. You can use this tag to control IS-IS route redistribution, route leaking, or route summarization by referencing it in a route map.
routing domain	A collection of connected areas that provide full connectivity to all end systems located within them. A routing domain is partitioned into areas.
system identifier	Uniquely identifies a system within an area
table map	A mechanism for applying a route map to an IS-IS route as a way to filter and manipulate route attributes before the route is added to the routing table.

Figure 18 illustrates some of the terms described in Table 13.

Figure 18: Overview of IS-IS Topology



ISO Network Layer Addresses

ISO network layer addresses are flexible enough to make routing feasible in a worldwide Internet. Network layer addresses in ISO and IP are hierarchical and clearly identify level 1 and level 2 areas. These addresses can be up to 20 octets long; any packet that contains an address has one additional octet to specify the length of the address.

An ISO address—also known as the NSAP address—is broken into three parts: the area address, the system identifier (ID), and the NSAP selector.

area address	system ID	selector

The area address defines the routing domain and the area within the routing domain. The length of the ID field can be from 1 to 8 octets and uses a single fixed length for any one routing domain. The selector field is always 1 octet long. Usually, all end systems within the same area have the same area address. Some areas can have multiple addresses. The NSAP address is defined by the network entity title (NET) during configuration.

Level 1 Routing

A level 1 router looks at a packet's area address and compares it with a destination address. If the area portion of the destination address matches its own area's address, the level 1 router uses the ID portion of the address to route the packet. If the area portion of the address does not match, the level 1 router routes the packet to a level 2 router within its area.

Level 2 Routing

Level 2 routers do not look at an area's internal structure, but simply route toward an area based on the area address. It is common for a level 2 router to also be a level 1 router in a particular area; these routers are sometimes referred to as level 1-2 routers. See Figure 18.

Dynamic Hostname Resolution

The system identifier of the NSAP address identifies a node in a network. System operators often find symbolic hostnames to be easier to use and remember than the system identifier. However, a static mapping of hostname to system identifier requires every router to maintain a table of the mappings; each table must contain the hostnames and system identifiers of every router in the network. The static mapping must be managed by router operators, and every change or addition of a mapping requires all the tables to be updated. Consequently, the static tables are likely to become rapidly outdated.

The router supports dynamic resolution of hostnames to system identifiers. You can use the **clns host** command to map the hostname to the NSAP address, and therefore to the system ID. This mapping is inserted in the dynamic hostname type-length-value tuple (TLV type 137), and subsequently advertised when LSPs are transmitted. The value field contains the hostname, preferably the fully qualified domain name (FQDN) of the host, or a subset of the FQDN. You can display the TLV by issuing the **show isis database detail** command.

Authentication

The router supports two authentication methods for IS-IS: simple authentication and hash function–based message authentication code (HMAC) MD5 authentication. These authentication methods prevent unauthorized routers from injecting false routing information into your network or forming adjacencies with your router.

By default, IS-IS authentication is disabled on the router until you enable it with the commands described in the following sections.

Simple Authentication

Simple authentication uses a text password (authentication key) that can be entered in encrypted or unencrypted form. The receiving router uses this authentication key to verify the packet.

You can configure the password for simple authentication by using the following commands:

- The **area-authentication-key** command assigns a password used by neighboring routers to authenticate IS-IS level 1 link-state PDUs (LSPs), complete sequence number PDUs (CSNPs), and partial sequence number PDUs (PSNPs). This command also enables simple authentication of level 1 LSPs.
- The **domain-authentication-key** command assigns a password used by neighboring routers to authenticate IS-IS level 2 LSPs, CSNPs, and PSNPs. This command also enables simple authentication of level 2 LSPs.

- The **isis authentication-key** command assigns a password associated with a specific interface for authentication of IS-IS level 1 or level 2 hello packets. This command also enables simple authentication of level 1 or level 2 hello packets.

These commands enable simple authentication of LSPs and (for the **isis authentication-key** command) hello packets only; they do not enable authentication of CSNP and PSNP packets. To enable authentication of CSNPs or PSNPs, you must issue either the **area-authentication** command or the **domain-authentication** command. For information, see *Enabling and Disabling Authentication of CSNPs and PSNPs* on page 317.

NOTE: The router supports simple authentication for compatibility with existing IS-IS implementations. However, we recommend that you do *not* use the simple authentication method because it is insecure (the text can be "sniffed").

HMAC MD5 Authentication

When you enable IS-IS HMAC MD5 authentication (also referred to as MD5 authentication), the router creates secure digests of the packets, encrypted according to the HMAC MD5 message-digest algorithms. The digests are inserted into the packets from which they are created. Depending on the commands you issue, the digests can be inserted into hello packets, link-state PDUs, complete sequence number PDUs, and partial sequence number PDUs.

You can configure an HMAC MD5 authentication key by using the following commands:

- The **area-message-digest-key** command specifies an HMAC MD5 key that the router uses to create a message digest of each level 1 packet—LSPs, CSNPs, and PSNPs—transmitted by area routers. Using MD5 authentication for area routers protects against unauthorized routers injecting false routing information into the area portions of your network. This command also enables MD5 authentication of level 1 LSPs.

- The **domain-message-digest-key** command specifies an HMAC MD5 key that the router uses to create a message digest of each level 2 packet—LSPs, CSNPs, and PSNPs—transmitted by domain routers. Using MD5 authentication for domain routers protects against unauthorized routers injecting false routing information into the routing domain portions of your network. This command also enables MD5 authentication of level 2 LSPs.

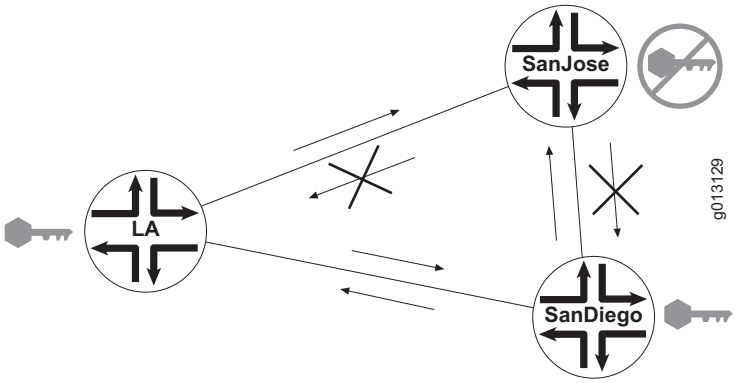
- The **isis message-digest-key** command specifies an HMAC MD5 key that the router uses to create a message digest of level 1 or level 2 hello packets on the interface. Level 1 packets are the default. Using MD5 authentication on interfaces protects against intrusion by preventing unauthorized routers from forming adjacencies with your router. This command also enables MD5 authentication of level 1 or level 2 hello packets.

These commands enable MD5 authentication of LSPs and (for the **isis message-digest-key** command) hello packets only; they do not enable authentication of CSNP and PSNP packets. To enable authentication of CSNPs or PSNPs, you must issue either the **area-authentication** command or the **domain-authentication** command. For information, see *Enabling and Disabling Authentication of CSNPs and PSNPs* on page 317.

MD5 Authentication Example

In the example shown in Figure 19, authentication is configured on router LA and router SanDiego, but not on router SanJose. Router LA and router SanDiego accept packets from each other because they contain message digests generated by an accepted key. Router SanJose accepts packets from router LA and router SanDiego, and simply ignores the message digest included in their packets. Router LA and router SanDiego reject packets from router SanJose because those packets do not include a message digest.

Figure 19: Packet Flow Between Routers With and Without Authentication Set



Specifying MD5 Start and Stop Timing

With each of the MD5 commands, you can specify when the router will start and stop *accepting* packets that include a digest made with this key. You can also specify when the router will start and stop *generating* packets that include a digest made with this key. If you specify a time for any of these actions, you can further specify the day, month, and year. The default times are as follows:

- Start accepting keys (startAcceptTime)—Current time
- Stop accepting keys (stopAcceptTime)—Never
- Start generating keys (startGenTime)—Current time plus 2 minutes
- Stop generating keys (stopGenTime)—Never

If you specify times, you must follow these guidelines to achieve appropriate timing between the actions:

- startAcceptTime must be less than startGenTime.
- stopGenTime must be less than stopAcceptTime.
- When a new key replaces an old one, the startGenTime time for the new key must be less than or equal to the stopGenTime time of the old key.

For example, suppose you configure authentication on router A and router B. If the startGenTime for router A is earlier than the startAcceptTime for router B, router B does not accept packets from router A until the current time matches its startAcceptTime.

The router accepts any packet authenticated with a key you have defined if the packet is received within the period defined for the key by its startAcceptTime and stopAcceptTime. If more than one key has been defined for that period, the router determines which key to use by comparing the startGenTime with the current time. When the startGenTime of a key matches the current time, the router starts using this key to transmit packets and stops using the previous key.

Example

The following commands configure both key 1 and key 2 to be accepted between 08:00:00 and 23:00:00. When the current time reaches 09:00:00, the router begins using key 1 to transmit packets. When the current time reaches 10:00:00, the router begins using key 2 to transmit packets; key 1 is no longer used. Key 2 will continue to be used until a new key is configured and the new key's startGenTime matches the current time on the router.

```
host1(config-router)#area-message-digest-key 1 hmac-md5 mr942s7n
start-accept 08:00:00 start-generate 9:00:00 stop-accept 23:00:00
stop-generate 22:59:59

host1(config-router)#area-message-digest-key 2 hmac-md5 dsb38h5f
start-accept 08:00:00 start-generate 10:00:00 stop-accept 23:00:00
stop-generate 22:59:59
```

Halting MD5 Authentication

To prevent key expiration from causing your network to revert to an unauthenticated condition, you cannot halt MD5 authentication by using the timers. When the stopGenTime time for a key is reached, the router does not stop generating the key if it was the last key issued. You must delete all keys to halt authentication. Use the **no** version of the command to delete a key.

Managing and Replacing MD5 Keys

A key has an infinite lifetime if you do not specify stopGenTime and stopAcceptTime. (As noted previously, if the last key expires, the router continues to generate that key.) Many system operators choose to change their keys on a regular basis, such as every month. If you determine that a key is no longer secure, configure a new key immediately. We recommend the following practice for configuring new keys:

1. Configure the new key on all routers in the IS-IS network.
2. Verify that the new key is working.
3. Delete the old key from every router.

Each key has an associated key-ID that you specify. The key-ID is sent with the message digest, so that the receiving routers know which key was used to generate the digest. You also use the key-ID to delete a key.

Enabling and Disabling Authentication of CSNPs and PSNPs

When the E-series router interoperates with other vendors' routers in the same network, you might want to enable or disable (suppress) authentication for some PDU types but not for others. For example, some vendors' routing software might not authenticate any PDUs, whereas other vendors' routing software might authenticate CSNPs and PSNPs separately from LSPs.

To facilitate interoperability with other vendors' routers, the E-series router allows you to enable and disable authentication of CSNPs and PSNPs separately from authentication of LSPs by using the following commands:

- The **area-authentication { csnp | psnp }** command enables or disables simple authentication or HMAC MD5 authentication of IS-IS level 1 CSNP packets or PSNP packets. By default, authentication of CSNPs and PSNPs is disabled.
- The **domain-authentication { csnp | psnp }** command enables or disables simple authentication or HMAC MD5 authentication of IS-IS level 2 CSNP packets or PSNP packets. By default, authentication of CSNPs and PSNPs is disabled.

When you suppress authentication of CSNPs, the router does not authenticate CSNP packets that it receives from neighboring routers, nor does it include authentication information in CSNP packets that it sends to other routers. Similarly, when you suppress authentication of PSNPs, the router neither authenticates PSNP packets that it receives nor sends authentication information in PSNP packets that it transmits.

Extensions for Traffic Engineering

The router supports *new-style* TLV tuples described in the Internet draft, *IS-IS Extensions for Traffic Engineering*. The router ID TLV (TLV type 134) contains the ID of the router that originates the LSP, providing a stable address that can always be referenced regardless of the state of node interfaces.

The extended IP reachability TLV (type 135) carries IP prefixes and is similar to the IP reachability TLVs (types 128 and 130). The extended IS reachability TLV (type 22) contains information about a series of IS neighbors and is similar to the IS neighbor TLV (type 2).

The older TLVs—2, 128, 130—each have a narrow metric field, providing for metric values ranging only from 0–63. The new TLVs—22 and 135—have a new data structure that includes a wide metric field of 3 bytes (extended IS reachability; configurable) or four bytes (extended IP reachability; calculated). Both new TLVs provide for the use of sub-TLVs to carry more information about IS neighbors; however, only the extended IS reachability TLV currently has defined sub-TLVs, such as IPv4 interface and neighbor addresses.

Use the **metric-style** commands to configure what style the router generates and accepts. The following behaviors are supported:

- Generates and accepts only old-style metrics
- Generates only old-style metrics, but accepts old style and new style
- Generates and accepts both old-style and new-style metrics (this option consumes the most system resources)
- Generates only new-style metrics, but accepts old style and new style
- Generates and accepts only new-style metrics

Refer to the Internet draft, *IS-IS Extensions for Traffic Engineering*, for more information about these extensions.

Integrated IS-IS

The E-series router supports the Integrated IS-IS version of IS-IS. Integrated IS-IS provides a single routing algorithm to route both TCP/IP and OSI Connectionless Network Protocol (CLNP) packets. This design adds IP-specific information to the OSI IS-IS routing protocol. It supports IP subnetting, variable subnet masks, type of service (ToS), and external routing.

Integrated IS-IS allows for the mixing of routing domains; that is, IP-only routers, OSI-only routers, and dual (IP and OSI) routers. OSI and IP packets are forwarded directly over the link-layer services without needing mutual encapsulation. The E-series router supports IS-IS only for the routing and forwarding of TCP/IP packets. Forwarding of OSI packets is not supported.

Equal-Cost Multipath

IS-IS supports equal-cost multipath (ECMP) and installs into the routing table multiple entries for paths to the same destination. Each of these multiple paths to a given destination must have the same cost as the others, but a different next hop.

Static PPP Interfaces

When IS-IS has been configured on a static PPP interface, the IS-IS neighbor does not come up if you remove the IP address from the interface and then add the IP address back to the interface. Consequently, when you remove and add back the IP address, you must also remove the IS-IS configuration from the interface and then add the configuration back to the interface by issuing the **no router isis** and **router isis** commands.

Route Tags

E-series routers support the use of route tags, also known as administrative tags, as a means of tagging the IP addresses on an IS-IS route before the route is propagated to other routers in an IS-IS domain. You must reference the tag in a route map to apply administrative policies to the IS-IS route that matches this tag.

Route Tag Applications

An administrative policy controls how a router handles the routes it receives from and sends to neighboring routers, and governs the installation of routes in the routing table. Examples of the types of administrative policies that you might apply with a route tag include:

- Policies for redistributing routes received from other protocols in the routing table to IS-IS
- Policies for redistributing routes between levels in an IS-IS routing hierarchy; this is also referred to as *route leaking*
- Policies for summarizing routes redistributed into IS-IS or within IS-IS by creating aggregate (summary) addresses

Route Tag Structure

On E-series routers, an IS-IS route tag is a 32-bit (4-octet) nonzero number that is stored as sub-TLV 1 inside the extended IP reachability TLV (type 135). TLV type 135, in turn, is part of an IS-IS LSP. The route tag is therefore advertised when LSPs are transmitted in an IS-IS network.

Because TLV type 135 is a new-style TLV tuple, it has a data structure that includes a wide metric field of four octets. As a result, to use IS-IS route tags you must issue the **metric-style wide** command (in Router Configuration mode) to specify that the router generate and accept only new-style TLV tuples.

For a discussion of IS-IS support for TLV tuples, see *Extensions for Traffic Engineering* on page 317.

Setting Route Tags

You can set IS-IS route tags in any of the following ways:

- Tagging a route for IP addresses on an IS-IS passive interface
- Tagging a route for IP addresses on an IS-IS interface
- Tagging IS-IS routes by using an associated route map to set the tag
- Tagging an IS-IS summary address

For instructions and examples on configuring IS-IS route tags, see the sections listed in Table 14.

Table 14: Configuration Tasks for Setting IS-IS Route Tags

To Learn About	Using This Command	See
Setting a route tag for an IS-IS passive interface	**passive-interface**	*Configuring Passive Interfaces* on page 337
Setting a route tag for an IS-IS interface	**isis tag**	*Configuring Route Tags for IS-IS Interfaces* on page 339
Setting a route tag for a route redistributed from another protocol to IS-IS by using an associated route map	**redistribute**	*Configuring Redistribution* on page 345
Setting a route tag for a route redistributed from one IS-IS level to another IS-IS level by using an associated route map	**redistribute isis ip**	*Redistributing Routes Between Levels* on page 347
Setting a route tag for an IS-IS default route by using an associated route map	**default-information originate**	*Configuring Default Routes* on page 352
Setting a route tag for an IS-IS summary address	**summary-address**	*Summarizing Routes* on page 354

Using Route Tags

You can set only a single route tag per IS-IS route. However, setting a tag for an IS-IS route has no effect by itself. To use the route tag to apply administrative policies such as route redistribution, route summarization, or route leaking, you must reference the tag value in a route map by issuing the **match tag** command (in Route Map Configuration mode). The route map must also include one or more **set** commands that modify attributes of the routes matching the tag value. These routes can reside on a different router than the one on which you set the route tag.

For example, the following commands define a route map to modify the metric and metric type attributes of IS-IS routes configured with a route tag value of 221. The **redistribute isis ip** command, as described in *Redistributing Routes Between Levels* on page 347, applies this route map when redistributing the routes from level 1 into level 2.

```
host1(config)#route-map map1 permit 5
host1(config-route-map)#match tag 221
host1(config-route-map)#set metric 10
host1(config-route-map)#set metric-type external
host1(config-route-map)#exit
host1(config)#router isis engineering
host1(config-router)#redistribute isis ip level-1 into level-2 route-map map1
```

Alternatively, you can use a route map to set the tag for an IS-IS route by issuing the **set tag** command (in Route Map Configuration mode). For example, the following commands define a route map that sets route tag 33 for those IS-IS routes configured with an administrative distance of 25:

```
host1(config)#route-map map2 permit 10
host1(config-route-map)#match distance 25
host1(config-route-map)#set tag 33
```

```
host1(config-route-map)#exit
host1(config)#router isis marketing
host1(config-router)#table-map map2
```

The **table-map** command, described in *Configuring Table Maps* on page 364, applies this route map to the IS-IS routes before they are added to the routing table. For details about configuring and using route maps, see *Route Maps* in *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

Unsupported Features

E-series routers do not currently support the following route tag features:

- Multiple route tags for a single IS-IS route

  Although the router accepts IS-IS routes with multiple route tags and propagates these routes in LSPs, it uses only the first route tag assigned to a route to determine routing policy.

- 64-bit (8-octet) route tags

  Although the router accepts IS-IS routes with 64-bit route tags and propagates these routes in LSPs, it does not use 64-bit route tags to determine routing policy.

- Mathematical (ordered) set operations on multiple route tags

Table Maps

E-series routers support the use of table maps to filter and manipulate the attributes of an IS-IS route before the route is installed in the routing table. Issuing the **table-map** command (in Router Configuration mode) applies a specified route map as a policy filter on the route before the route is installed in the routing table.

For IS-IS routes, the route map you apply by using the **table-map** command contains one or more **set** commands that can modify the following route attributes:

distance	origin
level	preference
metric	route type
metric type	tag

The router applies the specified route map to all routes currently and subsequently installed in the routing table. If any previously redistributed routes are changed as a result of applying the route map, the router redistributes these routes again with the changes caused by the route map.

For details about configuring and using route maps, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

Graceful Restart

E-series routers support IS-IS graceful restart as defined in RFC 3847—Restart Signaling for Intermediate System to Intermediate System (IS-IS) (July 2004). Graceful restart is also known as nonstop forwarding (NSF). When graceful restart is enabled on an IS-IS router, it allows the router to restart with minimal routing disruption to the network.

Features

When a router running in an IS-IS domain restarts, it typically causes routers in that domain to reset their adjacencies, thus generating unnecessary LSP flooding and shortest-path-first (SPF) calculations throughout the domain. Enabling graceful restart minimizes these effects by providing a mechanism by which a restarting router can do the following:

- Notify neighboring IS-IS routers that it is restarting and request help resynchronizing its LSP database. Neighbors with active adjacencies to the restarting router can thereby reestablish these adjacencies without having to reset them.
- Determine when complete LSP database synchronization with its neighbors has occurred.
- Optimize the process of LSP database synchronization while minimizing temporary routing disruption.

IS-IS graceful restart on E-series routers supports both restart and helper capabilities. These capabilities mean that an E-series router can not only notify neighboring IS-IS routers that it is restarting and request help resynchronizing its LSP database, but can also cooperate with other restarting routers to help them with the restart process.

How Graceful Restart Works

Graceful restart is disabled on the router by default. When you enable graceful restart by issuing the **nsf ietf** command, the router sends restart requests to neighboring routers to notify them that it is restarting. The restarting router includes the restart TLV (type 211) in its hello PDUs to signal the other routers that it supports graceful restart and to request help resynchronizing its LSP database. Including the restart TLV in hello packets also ensures that neighboring routers will maintain their active adjacencies to the restarting router and keep the restarting router in the network topology.

Graceful restart uses a set of configurable timers to support the restart mechanism. Table 15 briefly describes these timers and lists the associated commands that you can use to configure the timer values on the router.

Table 15: IS-IS Graceful Restart Timers

Timer	Description	Associated Command
Interface wait	Sets the maximum time (in seconds) that the router waits for all interfaces with IS-IS adjacencies to come up before completing the restart process	**nsf interface wait**
T1	Sets the time interval (in seconds) between restart requests sent by the router, and the number of times that the router resends unacknowledged restart requests	**nsf t1**
T2	Sets the maximum time (in seconds) that the router waits for the LSP database to synchronize	**nsf t2**
T3	Sets the maximum time (in seconds) that the restarting router waits before setting the overload bit to indicate that graceful restart has failed	**nsf t3**

For details about configuring graceful restart, see *Configuring Graceful Restart* on page 365.

IS-IS for IPv6

E-series routers support IPv6 routing for IS-IS. The IPv6 Reachability TLV propagates reachability information by flooding and is used in SPF calculations. The IPv6 Interface TLV is used for next hop calculation and is exchanged by means of IS-IS hello packets. A single SPF calculation computes both IPv6 and IPv4 routing tables.

IS-IS routers learn about their neighbors' support for IPv6 through the ISO network layer IPv6 protocol identifier, NLPID 142. The NLPID is contained in the NLPID TLV and is sent out in IS-IS hello packets when IS-IS IPv6 routing is enabled on an interface. A mismatch in support prevents an IS-IS adjacency from being established, because both neighbors must run the same protocols.

IPv6 aggregation, leaking, redistribution, export policies and import policies are supported similarly as for IP, but must be configured within the IS-IS IPv6 address family.

Graceful restart is supported for IS-IS IPv6 traffic depending on the availability of IPv6 high availability. It does not affect IP traffic.

Platform Considerations

For information about modules that support IS-IS on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IS-IS.

For information about modules that support IS-IS on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support IS-IS.

References

For more information about the IS-IS protocol, consult the following resources:

- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—See the Release Notes corresponding to your software release for information about maximum values.
- ISO International Standard 8473-1:1993—Information technology – Protocol for providing the connectionless-mode network service
- ISO International Standard 9542:1988 (E)—Information processing systems – Telecommunications and information exchange between systems – End System-to-Intermediate System Routing Exchange Protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)
- ISO/IEC 10589:1992—Information technology – Telecommunications and information exchange between systems – Intermediate System-to-Intermediate System Intra-Domain Routing Exchange Protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)
- Extended Ethernet Frame Size Support—draft-ietf-isis-ext-eth-01.txt (November 2001 expiration)
- Management Information Base for IS-IS—draft-ietf-isis-wg-mib-16.txt (January 2005 expiration)
- Point-to-point operation over LAN in link-state routing protocols—draft-ietf-isis-igp-p2p-over-lan-05.txt (January 2005 expiration)
- RFC 1195—Use of OSI IS-IS for Routing in TCP/IP and Dual Environments (December 1990)
- RFC 2763—Dynamic Hostname Exchange Mechanism for IS-IS (February 2000)
- RFC 2966—Domain-wide Prefix Distribution with Two-Level IS-IS (October 2000)
- RFC 2973—IS-IS Mesh Groups (October 2000)
- RFC 3277—Intermediate System to Intermediate System (IS-IS) Transient Blackhole Avoidance (April 2002)

- RFC 3373—Three-Way Handshake for Intermediate System to Intermediate System (IS-IS) Point-to-Point Adjacencies (September 2002)
- RFC 3784—Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE) (June 2004)
- RFC 3847—Restart Signaling for Intermediate System to Intermediate System (IS-IS) (July 2004)
- A Policy Control Mechanism in IS-IS Using Administrative Tags—draft-ietf-isis-admin-tags-02.txt (January 2005 expiration)

Features

Some of the major IS-IS features supported by the router include:

- Optimization of route leaking from level 1 to level 2
- Equal-cost paths maximum 16 equal paths
- Adjacency and LSP overrun
- Dynamic resolution of hostnames to system IDs
- Mesh groups
- Configurable LSP transmit and throttle intervals
- Route redistribution policies based on access lists between IS-IS levels
- Three-way handshake for point-to-point adjacencies
- Simple text and HMAC MD5 authentication
- Support for bigger metric TLVs
- Domain-wide prefix distribution
- Traffic engineering for MPLS
- 32-bit (4-octet) route tags
- Table maps
- Graceful restart
- IPv6 routing

Before You Run IS-IS

At least one IP address/router ID must be configured on your router for IS-IS to run.

Configuration Tasks

Configure Integrated IS-IS by completing the following tasks in the order presented. You must enable IS-IS. All other tasks are optional.

1. Enable IS-IS.
2. Configure selected IS-IS interface-specific parameters.
3. Configure selected global IS-IS parameters.
4. Configure selected IS-IS parameters for monitoring and debugging purposes.
5. Configure IS-IS parameters to enable CLNS packets to be recognized by your router and to monitor CLNS information.

Enabling IS-IS for IP Routing

When enabling IS-IS, you must create an IS-IS routing process and assign it to specific interfaces rather than to networks. You can specify only a single IS-IS process per router.

To enable IS-IS routing, enter Global Configuration mode, and follow this procedure:

1. Specify an IS-IS process for IP. In this example, floor12 is specified as the tag name.

   host1(config)#**router isis floor12**

   The router is now in Router Configuration mode.

2. Configure a Network Entity Title (NET) for the routing process that specifies the ISO network address.

   host1(config-router)#**net 47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00**

3. Enter Interface Configuration mode, and specify the interface that you want to actively route IS-IS.

   host1(config)#**interface atm 2/0**

4. Specify the IS-IS process to apply to the interface. Use the same tag name that you specified with the **router isis** command.

   host1(config-if)#**ip router isis floor12**

   You can repeat Steps 3 and 4 to apply the IS-IS process to multiple interfaces.

ip router isis

- Use to configure an IS-IS routing process on an IP interface.
- Before the IS-IS router process is useful, you must assign a NET with the **net** command, and enable some interfaces with IS-IS.

- Use the tag parameter to specify a meaningful name for a routing process. It must be unique among all IP routing processes for a given router. If you choose not to specify a tag name, a null tag is assumed, and the process is referenced with a null tag. Use the same tag name for **ip router isis** as you did for the **router isis** command.
- Example

  host1(config-if)#**ip router isis floor12**

- Use the **no** version to disable IS-IS for IP on the interface.

net

- Use to configure a NET for a specified routing process. The NET defines the ISO address and consists of an area address or ID, a system ID, and a selector.
- You must configure a minimum of one NET.
- You can have a maximum of three NETs per router.
- You can manually add multiple area IDs by adding multiple NETs with the same system ID.
- There is no default value; **net** must be configured for an IS-IS process to start.
- Multiple NETs can be temporarily useful when there has been a network reconfiguration where either multiple areas are merged, or one area is in the process of being split into more areas. Multiple area addresses enable you to renumber an area slowly, without needing to set aside time to renumber areas all at once.
- When you use IS-IS to do IP routing only, a NET must be configured to instruct the router about its system ID and area ID.
- Example—The following commands configure a router with the area ID 47.0005.80ff.f800.0000.0001.0001 and the system ID 0000.0c11.1111. The last byte of the NET is the N-selector byte and is always 0.

  host1(config-router)#**net**
  47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00

- Use the **no** version to remove a specific NET. Remember that you must specify the NET. The last NET cannot be removed.

router isis

- Use to enable the IS-IS routing protocol and to specify an IS-IS process for IP.
- Specify only one IS-IS process per router.
- Use the tag parameter to specify a meaningful name for a routing process. If you choose not to specify a tag name, a null tag is assumed, and the process is referenced with a null tag.
- Example

  host1(config)#**router isis floor12**

- Use the **no** version to disable IS-IS routing.

Summary Example

```
host1(config)#router isis floor12
host1(config-router)#net 47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00
host1(config-router)#exit
host1(config)#interface atm 2/0
host1(config-if)#ip router isis floor12
host1(config-router)#exit
host1(config-if)#interface atm 2/1
host1(config-if)#ip router isis floor12
```

Enabling and Configuring IS-IS for IPv6 Routing

When enabling IS-IS IPv6, you must create an IS-IS IPv6 routing process and assign it to specific interfaces rather than to networks. You can specify only a single IS-IS process per router.

To enable IS-IS routing, enter Global Configuration mode, and follow this procedure:

1. Access Global Configuration mode and specify an IPv6 license.

   host1(config)#**license ipv6** *license-value*

2. Configure an IP address on the router to serve as the router ID.

   host1(config)#**interface loopback0**
   host1(config-if)#**ip address 10.6.5.4/32**

3. Configure the lower-layer interfaces over which the IPv6 traffic flows.

   host1(config-if)#**interface fastEthernet 1/0**

4. Configure an IPv6 address on the interface.

   host1(config-if)#**ipv6 address 2008::1/48**

5. Specify the IS-IS IPv6 process to apply to the interface. Use the same tag name that you specify with the **router isis** command for the VR.

   host1(config-if)#**ipv6 router isis floor12**

   Repeat Steps 3–5 for all desired IPv6 interfaces.

6. Specify an IS-IS process globally for the VR. Use the same tag name that you specify with the **ipv6 router isis** command on the interface.

   host1(config)#**router isis floor12**

7. Configure a Network Entity Title (NET) for the routing process that specifies the ISO network address.

   host1(config-router)#**net**
   47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00

8. Create the IS-IS IPv6 address family for the interface.

   host1(config-router)#**address-family ipv6 unicast**

9. Configure any of the following desired IS-IS options for the address family: redistributing routes from other protocols, redistributing IS-IS IPv6 routes between levels, distributing level 2 routing information to level 1 routers throughout the IS-IS routing domain, summarizing IPv6 routes, applying a route map to modify routes before they are installed in the routing table,

   host1(config-router-af)#**redistribute ospf level-1-2**
   host1(config-router-af)#**redistribute isis level-2 into level-1**
   host1(config-router-af)#**distribute-domain-wide**
   host1(config-router-af)#**summary-prefix 2001:2000::0/8 level-1 metric 10 tag 100**
   host1(config-router-af)#**table-map ospfFilter**

10. Exit the IS-IS IPv6 address family.

    host1(config-router-af)#**exit-address-family**

NOTE: Enabling IPv6 for the interface also enables IPv4 for that interface. However, this interface does not participate in IS-IS IPv4 routing.

address-family

- Use to configure IS-IS to exchange IPv6 addresses by creating the IPv6 address family.
- Use the **unicast** keyword to exchange unicast addresses. Use the **multicast** keyword to exchange multicast addresses. Use the **unicast** and **multicast** keywords together, or omit both of them to exchange both unicast and multicast addresses.
- Examples

  host1(config)#**address-family ipv6 unicast**

- Use the **no** version to disable the exchange of IPv6 addresses.

exit-address-family

- Use to exit Address Family Configuration mode and access Router Configuration mode.
- Example

  host1:vr1(config-router-af)#**exit-address-family**

- There is no **no** version.

ipv6 router isis

- Use to configure an IS-IS routing process on an IPv6 interface.
- Before the IS-IS router process is useful, you must assign a NET with the **net** command, and enable some interfaces with IS-IS.
- Use the tag parameter to specify a meaningful name for a routing process. It must be unique among all IPv6 routing processes for a given router. If you choose not to specify a tag name, a null tag is assumed, and the process is referenced with a null tag. Use the same tag name for **ipv6 router isis** as you did for the **router isis** command.
- Example—Enables ISIS for IPv6 on an interface.

  host1(config-if)#**ipv6 router isis bldg1**

- Use the **no** version to disable IS-IS on the interface.

Summary Example

```
host1(config)#license ipv6 license-value
host1(config)#interface loopback0
host1(config-if)#ip address 10.6.5.4/32
host1(config-if)#interface fastEthernet 1/0
host1(config-if)#ipv6 address 2008::1/48
host1(config-if)#ipv6 router isis floor12
host1(config)#router isis floor12
host1(config-router)#net
47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00
host1(config-router)#address-family ipv6 unicast
host1(config-router-af)#redistribute ospf level-1-2
host1(config-router-af)#redistribute isis level-2 into level-1
host1(config-router-af)#distribute-domain-wide
host1(config-router-af)#summary-prefix 2001:2000::0/8 level-1 metric 10 tag
100
host1(config-router-af)#table-map ospfFilter
```

Configuring IS-IS Interface-Specific Parameters

You can change IS-IS interface–specific parameters; most can be configured independently of other attached routers. You are not required to alter any interface parameters; however, some parameters must be consistent across all routers in your network. If you change certain values from the defaults, you must configure them on multiple interfaces and routers.

In the following command guidelines, many parameters are preset to a default value. If that parameter has been modified from its default, use the **no** version of the command to restore its default value.

Configuring Authentication

You can set a password to authenticate IS-IS hello packets, and you can configure HMAC MD5 authentication for IS-IS interfaces.

isis authentication-key

- Use to specify a password associated with an interface for authentication of IS-IS hello packets, and to enable simple authentication of level 1 or level 2 hello packets.
- You can specify whether the password is for level 1 or level 2 hellos.
- Example

  host1(config-if)#**isis authentication-key 0 red5flower6**

- Use the **no** version to delete the password.

isis message-digest-key

- Use to configure HMAC MD5 authentication for an interface, and to enable MD5 authentication of level 1 or level 2 hello packets.
- Generates a secure, encrypted message digest of level 1 or level 2 hello packets and inserts the digest into the packet from which it is created. Level 1 is the default.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-if)#**isis message-digest-key 3 hmac-md5 wdi6c3s39n level-2**

- For point-to-point interfaces, configure keys only for level 1, because only one hello packet is sent (at level 1), not one at level 1 and one at level 2. Keys configured at level 2 are ignored for point-to-point interfaces.
- Use the **no** version to delete the MD5 key, specified by the key ID, from the interface.

Configuring Link-State Metrics

You can configure the routing metric (cost) for an IS-IS interface. Routes with lower total path metrics are preferred over those with higher path metrics.

isis metric

- Use to configure a cost for a specified interface.
- You can select a number in the range 0–63 if you configured the router with the **metric-style narrow** command. You can select a number in the range 0–16277215 if you configured the router with the **metric-style transition** or the **metric-style wide** command.
- The default value is 10. The default metric is the value assigned when no quality of service (QoS) routing is performed.
- You can configure the default metric for a specified interface by selecting level 1 or level 2 routing. This resets the metric only for level 1 or level 2 routing, respectively. If you do not specify a level, the command specifies both level 1 and level 2 by default.
- We recommend that you configure a reference bandwidth if you want the default cost on interfaces to be related to link speed. If you do not, the default IS-IS metrics are simply hop-count-like metrics.
- Example

  host1(config-if)#**isis metric 20 level-2**

- Use the **no** version to restore the default value, 10.

Configuring a Reference Bandwidth to Set a Default Metric

By default, all IS-IS interfaces without a configured metric have the same routing metric, 10, However, when you configure a reference bandwidth for IS-IS, the default metric is calculated differently for each IS-IS interface. The default routing metric in this case is the reference bandwidth divided by the bandwidth of the particular interface.

For example, if you set the IS-IS reference bandwidth to 50,000,000, the default metric for a 10-Mbps interface is calculated as 5. Interfaces with lower bandwidths have higher default metrics than this interface. Similarly, links with higher bandwidths have lower default metrics than this interface.

reference-bandwidth

- Use to set a reference bandwidth from which a default metric can be calculated by IS-IS for interfaces without a configured metric.
- Example

  host1(config-router)#**reference-bandwidth 100000000**

- Use the **no** version to remove the reference bandwidth. When you do so, the default metric reverts to 10.

Setting the CSNP Interval

You can set the advertised complete sequence number PDU (CSNP) interval for an IS-IS interface.

isis csnp-interval

- Use to configure the **isis csnp-interval** level for a specified interface. The level can be configured independently for level 1 and level 2.
- For LAN interfaces: the default value is 10 seconds, which you probably do not need to change. For WAN interfaces: the default value is 0 seconds or disabled.
- On point-to-point subinterfaces use **isis csnp-interval** with the **isis mesh-group** command.
- Completed sequence number PDUs are sent by the designated router to maintain database synchronization.
- Example

  host1(config-if)#**isis csnp-interval 30 level-1**

- Use the **no** version to restore the default value.

Configuring Hello Packet Parameters

You can set the hello interval and the hello multiplier for IS-IS hello packets.

isis hello-interval
isis hello-multiplier

- Use the **isis hello-interval** command to set the length of time (in seconds) between hello packets sent on a specific interface. Configure independently for level 1 and level 2, except on point-to-point interfaces because only a single type of hello packet is sent on serial links. For this reason, it is independent of levels 1 and 2. For example, you can specify an optional level for Frame Relay multiaccess networks.

  The hello-interval is equal to the *hello multiplier* times the *hello interval seconds* and is advertised as the *holdtime* in the hello packets transmitted. The range is 0–65535; the default value is 10 seconds.

NOTE: The hello-interval value must be the same for all routers attached to a common network. With smaller hello intervals, topological changes are detected faster, but there is more routing traffic.

- Use the **isis hello-multiplier** command to set a number by which to multiply the hello interval seconds. This number determines the total *holding time* transmitted in the IS-IS hello packet. The default is 3. Use when hello packets are frequently lost and IS-IS adjacencies are failing unnecessarily.

  The advertised hold time in IS-IS hellos is set to the hello-multiplier times the hello-interval. Neighbors declare an adjacency to this router to be down after not having received any IS-IS hellos during the advertised hold time.

  - The hold time (and thus the hello-multiplier and the hello-interval) can be set on a per interface basis, and can be different between different routers in one area.
  - Using a smaller hello-multiplier will give fast convergence, but can result in more routing instability.
  - Increment the hello-multiplier to a larger value to help network stability when needed.

CAUTION: Never configure a hello-multiplier lower than the default.

- Holding time—Time a neighbor waits for another hello packet before declaring the neighbor is down. It determines how quickly a failed link or neighbor is identified so that routes can be recalculated.

- Raise the hello multiplier and lower the hello interval simultaneously to make the hello protocol more reliable without increasing the time required to detect a link failure.
- Example

  host1(config-if)#**isis hello-interval 6 level-1**
  host1(config-if)#**isis hello-multiplier 10 level-1**

- Use the **no** version to restore a default value.

Padding IS-IS Hello Packets

You can use the **isis hello padding** command to configure IS-IS hello packet padding. Padding the hello packets promotes early error detection due to transmission problems with large frames or due to mismatched MTUs on adjacent interfaces.

When disabled (default), IS-IS hello packets are padded to the full MTU size until an adjacency is formed with the adjacent interface. After the adjacency is formed, the hello packets are no longer padded. When enabled, IS-IS hello packets are always padded.

isis hello padding

- Use to pad IS-IS hello packets to their full maximum transmission unit (MTU) size.
- Example

  host1(config-if)#**isis hello padding**

- Use the **no** version to restore the hello padding to its default, no padding.

Configuring LSP Parameters

You can configure the transmission interval, retransmission interval, and retransmission throttle interval for LSPs on an interface-specific basis.

isis lsp-interval

- Use to configure the delay between successive IS-IS link-state PDU (LSP) transmissions.
- You can choose an interval in the range 1–4294967295 milliseconds. For example, setting 100 milliseconds allows 10 packets per second.
- The default value is 33 milliseconds.
- If your network has many IS-IS neighbors and interfaces, a particular router may have difficulty with the CPU load imposed by LSP transmission and reception. If this is the case, you can reduce the LSP transmission rate by issuing this command.
- Example

  host1(config-if)#**isis lsp-interval 100**

- Use the **no** version to restore the default value, 33 milliseconds.

isis retransmit-interval

- Use to configure the number of seconds between the retransmission of IS-IS LSPs with the same LSP ID for point-to-point links.
- You can select an interval in the range 1–65535 seconds.
- The default value is 5 seconds.
- Specify a number greater than the expected round-trip delay between any two routers on your network.
- Always specify conservatively; otherwise, excessive retransmission can result.
- Because retransmissions occur only when LSPs are dropped, when you set **isis retransmit-interval** to a higher value, it has little effect on reconvergence.
- Set to a higher value when routers have many neighbors or more paths over which LSPs can be flooded.
- Use a large value for serial lines.
- Example

  host1(config-if)#**isis retransmit-interval 60**

- Use the **no** version to restore the default value, 5 seconds.

isis retransmit-throttle-interval

- Use to configure the maximum rate at which IS-IS LSPs are retransmitted on point-to-point links. The interval is the number of milliseconds between packets.
- You can choose an interval in the range 0–65535 milliseconds.
- The default delay value is 33 milliseconds.
- The **isis retransmit-throttle-interval** is the maximum rate at which IS-IS LSPs are retransmitted. It is different from **isis lsp-interval**, which is the rate at which LSPs are transmitted on the interface; and it is different from **isis retransmit-interval**, which is the period between successive retransmissions of the *same* LSP. Use all three commands with each other to control the load of routing traffic from one router to its neighbors.
- Typically, you can set this interval for very large networks with many LSPs and many interfaces as a way of controlling LSP retransmission traffic.
- Example

  host1(config-if)#**isis retransmit-throttle-interval 300**

- Use the **no** version to restore the default value, 33 milliseconds.

Setting the Designated Router Priority

You can set the priority for the designated IS-IS router that you have elected to use.

isis priority

- Use to set the priority of use for your designated router.
- You can configure an individual priority for level 1 and level 2 by choosing a priority level in the range 0–127.
- The default priority level is 64.
- Specifying the **level 1** or **level 2** keyword resets the priority only for level 1 or level 2 routing, respectively.
- Priorities are used to determine which router in the network is the designated intermediate system (DIS); the router with the highest priority becomes the DIS. Priorities are advertised in hellos.
- IS-IS has no backup designated router. Setting the priority to 0 reduces the chance of this router becoming the DIS, but does not prevent it. If a router with a higher priority is identified, it takes over the role from the current DIS. When priorities are equal, the highest MAC address breaks the tie and becomes the DIS.
- Example

  host1(config-if)#**isis priority 80 level-1**

- Use the **no** version to restore the default value, 64.

Configuring Passive Interfaces

You can configure an IS-IS passive interface. A passive interface only advertises its IP address in its LSPs; it does not send or receive IS-IS packets.

Optionally, you can set a route tag for an IS-IS passive interface by including the **tag** keyword and a numeric tag value in the **passive-interface** command.

Passive interfaces have a metric of zero by default. You can set a different metric for a particular passive interface by specifying the value along with the **metric** keyword. A global default metric set with the **metric** command does not affect any passive interface. Similarly, configuring a reference bandwidth for IS-IS has no effect on passive interfaces. Metrics specified for a passive interface apply to both level 1 and level 2 interfaces unless you restrict the metric to a single level.

passive-interface

- Use to configure an IS-IS interface so that its IP address is advertised in its link-state PDUs but no IS-IS packets are sent from or received on the interface.
- Use the optional **tag** keyword to specify a tag value for an IS-IS passive interface before the route is propagated to other routers in an IS-IS domain. The tag value must be a number in the range 1–4294967295.
- Use the optional **metric** keyword to specify a metric value for an IS-IS passive interface. The metric value must be a number in the range 1–16777215. This value overrides the default metric of zero.

- You can also accomplish the equivalent of the **passive-interface** command by using the **redistribute** command to redistribute a connected route to level 1.
- Example 1—Configures loopback 0 as a passive interface and enable IS-IS on subinterfaces ATM 2/0.1 and ATM 2/1.1. IS-IS advertises the IP address of loopback 0 in its link-state PDUs, but runs only on ATM 2/0.1 and ATM 2/1.1:

  host1(config)#**router isis floor12**
  host1(config-router)#**net**
  47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00
  host1(config-router)#**passive-interface loopback 0**
  host1(config-router)#**exit**
  host1(config)#**interface atm 2/0.1**
  host1(config-subif)#**ip router isis floor12**
  host1(config-subif)#**exit**
  host1(config)#**interface atm 2/1.1**
  host1(config-subif)#**ip router isis floor12**

  You can override the passive-interface configuration simply by issuing the complementary command. For example, suppose you issue the following commands after the previous configuration:

  host1(config-router)#**passive-interface atm 2/0.1**
  host1(config-router)#**exit**
  host1(config)#**interface loopback 0**
  host1(config-if)#**ip router isis floor12**

  Now IS-IS advertises the IP address of ATM 2/0.1 in its link-state PDUs, but runs only on loopback 0 and ATM 2/1.1.

- Example 2—Sets a route tag on the IS-IS passive interface configured in Example 1.

  host1(config)#**router isis floor12**
  host1(config-router)#**passive-interface loopback 0 tag 12**

- Example 3—Sets a metric and level on the IS-IS passive interface configured in Example 1.

  host1(config)#**router isis floor12**
  host1(config-router)#**passive-interface loopback 0 metric 45 level-2**

- Use the **no** version to delete the passive interface, or to remove the tag, metric, or both.

Configuring Adjacency

You can configure the type (level) of adjacency you want to use on an IS-IS interface.

isis circuit-type

- Use to specify adjacency levels on a specified interface; however, normally, you do not need to use this command.
- Configure a router as a level 1-only, a level 1–level 2 system, or a level 2-only system.
- Configure some interfaces to be level 2-only for routers that are between areas. This prevents wasting bandwidth by sending out unused level 1 hellos.

- On point-to-point interfaces, the level 1 and level 2 hellos are in the same packet.
- Level 1-2 is the default.
- Example

  host1(config-if)#**isis circuit-type level-2-only**

- Use the **no** version to restore the default value, level-1-2.

Configuring Route Tags for IS-IS Interfaces

To configure a route tag for the IP addresses on an IS-IS interface:

1. Specify an IS-IS routing process, and access Router Configuration mode.

   host1(config)#**router isis engineering**
   host1(config-router)#

2. Configure a NET for the IS-IS process.

   host1(config-router)#**net**
   47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00

3. Configure the router to accept and generate only new-style TLV tuples with a wider metric field. New-style TLV tuples include TLV type 135, which contains the route tag.

   host1(config-router)#**metric-style wide**

4. Exit Router Configuration mode.

   host1(config-router)#**exit**

5. Specify the interface on which you want to route IS-IS.

   The procedure assumes that at least one IP address is already configured on this interface.

   host1(config)#**interface atm 2/2.1**

6. Configure a route tag for the interface.

   host1(config-subif)#**isis tag 221**

7. Specify the IS-IS process to apply to the interface.

   host1(config-subif)#**ip router isis engineering**

8. (Optional) Access Privileged Exec mode, and verify the route tag assignment.

   host1(config-subif)#**exit**
   host1(config)#**exit**
   host1#**show isis database detail**

isis tag

- Use to set a route tag for the IP addresses on an IS-IS interface before the route is propagated to other routers in an IS-IS domain.
- Specify a numeric tag value in the range 1–4294967295.
- To make use of the route tag to modify route attributes or redistribute routes, you must reference the tag value in a route map.
- Example

  host1(config)#**interface atm 3/0**
  host1(config-if)#**isis tag 45**

- Use the **no** version to remove the route tag from the interface.

Configuring Point-to-Point-over-LAN Circuits

You can deploy IS-IS on broadcast and point-to-point circuits. IS-IS treats these circuits differently in several ways, such as when establishing neighbor adjacencies or flooding link-state information.

Broadcast circuits use designated routers and are represented as virtual nodes in the network topology. They require periodic database synchronization. By default, IS-IS treats the broadcast link as LAN media and tries to bring up the LAN adjacency even when the interface is configured as unnumbered or only a single neighbor exists on that link.

In contrast, point-to-point circuits have less overhead, because they do not use designated routers, the link-state database has no representation of the pseudonode or network LSA, and they do not require periodic database synchronization. However, if more than two routers are connected on the LAN media, routing information in the network is reduced.

Although broadcast circuits are intended to handle more than two devices, in some circumstances you might connect only two routers over the physical or virtual LAN. Even though only two routers are connected, IS-IS treats the circuit as a broadcast circuit that has many more connected routers, with all the associated broadcast overhead but without the benefits of reduced routing information and of optimized flooding that result from having more than two routers on the LAN.

You can use the **isis network point-to-point** command to configure IS-IS to operate using point-to-point connections on a broadcast circuit when only two routers are on the circuit. This configuration is known as a point-to-point-over-LAN or P2P circuit. This interface configuration tears down the current LAN adjacency that IS-IS has over this interface. IS-IS then reestablishes the adjacency as a point-to-point connection and regenerates the LSPs. The broadcast link is thereafter treated as simple point-to-point interface.

Treating the LAN as a P2P circuit reduces the amount of information that IS-IS has to maintain and manage. For example, there is no need to elect a designated router for the interface. LSP flooding is performed as in P2P links without the need for using periodic CSNPs.

This circuit configuration can be advantageous even when many routers are on the LAN. For example, you might want to organize the routers into multiple smaller VLANs so that you can assign different costs to the IS-IS neighbors. You can apply this configuration to any such VLAN that has only two routers. IS-IS then views the LAN as a mesh of point-to-point connections.

The use of IP unnumbered interfaces makes the most of scarce IP address resources and provides for simpler network management and configuration. This configuration enables IP processing on a point-to-point interface without an explicit IP address. The IP unnumbered interface borrows the IP address of another interface on the node. Point-to-point-over-LAN circuits separate the concept of network type from media type, and enable you to apply unnumbered interface configurations to LANs.

The point-to-point-over-LAN feature requires the following:

- The LAN must have only two routers.
- Both routers must support the feature.
- You must configure the interface at each end as a P2P connection.
- If you are using numbered interfaces, both ends must be in same IPv4 subnet.
- If you are using unnumbered interfaces, both ends require static ARP entry configuration.

isis network point-to-point

- Use to specify that the broadcast circuit is to be treated as a point-to-point circuit.
- Issuing this command tears down existing adjacencies, originates or flushes LSPs, and establishes new adjacencies
- Example

  host1(config-intf)#**isis network point-to-point**

- Use the **no** version to restore the default value, treating the circuit as a broadcast circuit.

Summary Example

```
host1(config-router)#passive-interface loopback 0
host1(config-if)#interface atm 8/0
host1(config-if)#isis tag 55
host1(config-if)#isis metric 20 level-2
host1(config-if)#isis csnp-interval 30 level-1
host1(config-if)#isis hello-interval 6 level-1
host1(config-if)#isis hello-multiplier 10 level-1
host1(config-if)#isis lsp-interval 100
host1(config-if)#isis retransmit-interval 60
host1(config-if)#isis retransmit-throttle-interval 300
host1(config-if)#isis priority 80 level-1
host1(config-if)#isis circuit-type level-2-only
host1(config-intf)#no isis network point-to-point
```

Configuring Global IS-IS Parameters

This section describes the commands you can use to globally configure optional IS-IS parameters.

In the following command guidelines, many parameters are preset to a default value. Use the **no** version of those commands to restore default values.

Setting Authentication Passwords

You can configure simple authentication or HMAC MD5 authentication for either an area or a domain.

area-authentication-key

- Use to specify a password used by neighboring routers for authentication of IS-IS level 1 LSPs, CSNPs, and PSNPs.
- Issuing this command enables simple authentication of level 1 LSPs only. To enable simple authentication of level 1 CSNPs or PSNPs, use the **area-authentication** command, described on page 344.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**area-authentication-key 0 bigtree**

- Use the **no** version to delete the password.

area-message-digest-key

- Use to configure HMAC MD5 authentication for an area.
- Generates a secure, encrypted message digest of level 1 packets (LSPs, CSNPs, and PSNPs) and inserts the digest into the packet from which it is created.
- Issuing this command enables MD5 authentication of level 1 LSPs only. To enable MD5 authentication of level 1 CSNPs or PSNPs, use the **area-authentication** command, described on page 344.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**area-message-digest-key 1 hmac-md5 kd4s8hnEK**

- Use the **no** version to delete the MD5 key specified by the key ID.

domain-authentication-key

- Use to specify a password used by neighboring routers for authentication of IS-IS level 2 LSPs, CSNPs, and PSNPs.
- Issuing this command enables simple authentication of level 2 LSPs only. To enable simple authentication of level 2 CSNPs or PSNPs, use the **domain-authentication** command, described on page 344.

- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**domain-authentication-key 8 4kl6n39us**

- Use the **no** version to delete the password.

domain-message-digest-key

- Use to configure HMAC MD5 authentication for a domain.
- Generates a secure, encrypted message digest of level 2 packets (LSPs, CSNPs, and PSNPs) and inserts the digest into the packet from which it is created.
- Issuing this command enables MD5 authentication of level 2 LSPs only. To enable MD5 authentication of level 2 CSNPs or PSNPs, use the **domain-authentication** command, described on page 344.
- You can specify whether the key is entered in unencrypted or encrypted format. If you do not specify which, the string is assumed to be unencrypted.
- Example

  host1(config-router)#**domain-message-digest-key 4 hmac-md5 4bFjt7es**

- Use the **no** version to delete the MD5 key specified by the key ID.

Configuring Authentication of CSNPs and PSNPs

You must enable and disable authentication of CSNP packets and PSNP packets separately from authentication of LSP packets.

area-authentication

- Use to enable or disable (suppress) simple authentication or HMAC MD5 authentication of IS-IS level 1 CSNP packets or PSNP packets.
- When authentication is enabled, it uses either the simple text password specified by the **area-authentication-key** command, or the HMAC MD5 key specified by the **area-message-digest-key** command.
- You must specify either the **csnp** keyword to enable authentication of level 1 CSNP packets, or the **psnp** keyword to enable authentication of level 1 PSNP packets.
- Example

  host1(config-router)#**area-authentication csnp**

- Use the **no** version to restore the default behavior, in which authentication of level 1 CSNPs and PSNPs is disabled. When authentication of level 1 CSNPs or PSNPs is suppressed, the router does not authenticate these packets when it receives them, nor does it send authentication information in these packets when it transmits them.

domain-authentication

- Use to enable or disable (suppress) simple authentication or HMAC MD5 authentication of IS-IS level 2 CSNP packets or PSNP packets.
- When authentication is enabled, it uses either the simple text password specified by the **domain-authentication-key** command, or the HMAC MD5 key specified by the **domain-message-digest-key** command.
- You must specify either the **csnp** keyword to enable authentication of level 2 CSNP packets, or the **psnp** keyword to enable authentication of level 2 PSNP packets.
- Example

  host1(config-router)#**domain-authentication csnp**

- Use the **no** version to restore the default behavior, in which authentication of level 2 CSNPs and PSNPs is disabled. When authentication of level 2 CSNPs or PSNPs is suppressed, the router does not authenticate these packets when it receives them, nor does it send authentication information in these packets when it transmits them.

Configuring Redistribution

You can specify how IS-IS redistributes routes received from other routing protocols, redistributes routes according to new policies, and controls redistribution of routes with access lists and route maps.

Optionally, when you issue the **redistribute** command and specify a route map, you can use the map to set a route tag for a route redistributed from another protocol to IS-IS. Make sure the route map you specify includes the **set tag** command that defines a tag value for the routes destined for IS-IS. For details about configuring and using route maps, see *Route Maps* in *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

To redistribute IPv6 routes, issue the **redistribute** command from within the IS-IS IPv6 address family.

access-list
route-map

- Use the **access-list** command to create a standard or extended access list.
- Use the **route-map** command to create a route map.
- For detailed information about configuring access lists and route maps, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.
- Example—For IP route redistribution the access list filters IP routes; for IPv6 route redistribution, the access list must filter IPv6 routes.

  1. Configure three static routes:

     host1(config)#**ip route 10.20.20.0 255.255.255.0 192.168.1.0**
     host1(config)#**ip route 10.20.21.0 255.255.255.0 192.168.1.0**
     host1(config)#**ip route 10.21.0.0 255.255.255.0 192.168.1.0**

  2. Configure an access list with filters on routes 10.20.20.0/24 and 10.20.21.0/24:

     host1(config)#**access-list boston permit 10.20.0.0 0.0.255.255**

  3. Configure a route map that matches the previous access list and applies an internal metric type:

     host1(config)#**route-map 1**
     host1(config-route-map)#**match ip address 1**
     host1(config-route-map)#**set metric-type internal**

  4. Configure redistribution into IS-IS of the static routes with route map 1:

     host1(config)#**router isis testnet**
     host1(config-router)#**redistribute static ip route-map 1**

5. Use the **show isis database** command to verify the effect of the redistribution (that two static routes matching the route map are redistributed as level 2 internal routes):

```
host1#show isis database detail l2
IS-IS Level-2 Link State Database
LSPID LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0000.6666.00-00  0x000002B7   0x3E1F 1198 0/0/0
  Area Address: 47.0005.80FF.F800.0000.0001.0001
  NLPID:       0xcc
  IP Address:  192.168.1.105
  Metric: 10 IS 0000.0000.6666.01
  Metric: 10 IS 0000.0000.3333.00
  Metric: 10 IS 0000.0000.7777.00
  Metric: 30 IP 10.20.21.0 255.255.255.0
  Metric: 30 IP 10.20.20.0 255.255.255.0
```

- Use the **no** version of the **access-list** command to remove the access list or the specified entry in the access list.
- Use the **no** version of the **route-map** command to remove an entry.

clear ip isis redistribution
clear isis ipv6 redistribution

- Use to clear all the routes that have been previously redistributed into IS-IS and to redistribute them using the current policy configured. Use the IP version to redistribute IP routes. Use the IPv6 version to redistribute IPv6 routes.
- Use when you have made changes to route maps or access lists that affect how routes are redistributed to IS-IS.
- Example

  host1#**clear ip isis redistribution**

- There is no **no** version.

disable-dynamic-redistribute

- Use to halt the dynamic redistribution of routes that are initiated by changes to a route map.
- Dynamic redistribution is enabled by default.
- Example

  host1(config-router)#**disable-dynamic-redistribute**

- Use the **no** version to reenable dynamic redistribution.

redistribute

- Use to redistribute routes from other routing protocols in the routing table to IS-IS. IS-IS advertises these routes as level 1 only, level 2 only, or both. Level 2 only is the default.
- To redistribute IPv6 routes, you must issue the command from within the IS-IS IPv6 address family.
- The default is no source protocol defined for redistribution.

- This command can accomplish the same results as the **passive-interface** command by redistributing a connected route to level 1.
- Optionally, you can specify a route map and use it to set a route tag for routes redistributed to IS-IS.
- Example 1—Redistributing static IP routes with a route map

  host1(config-router)#**redistribute static ip route-map 10**

- Example 2—Redistributing IPv6 routes from OSPF into IS-IS level 1 and level 2

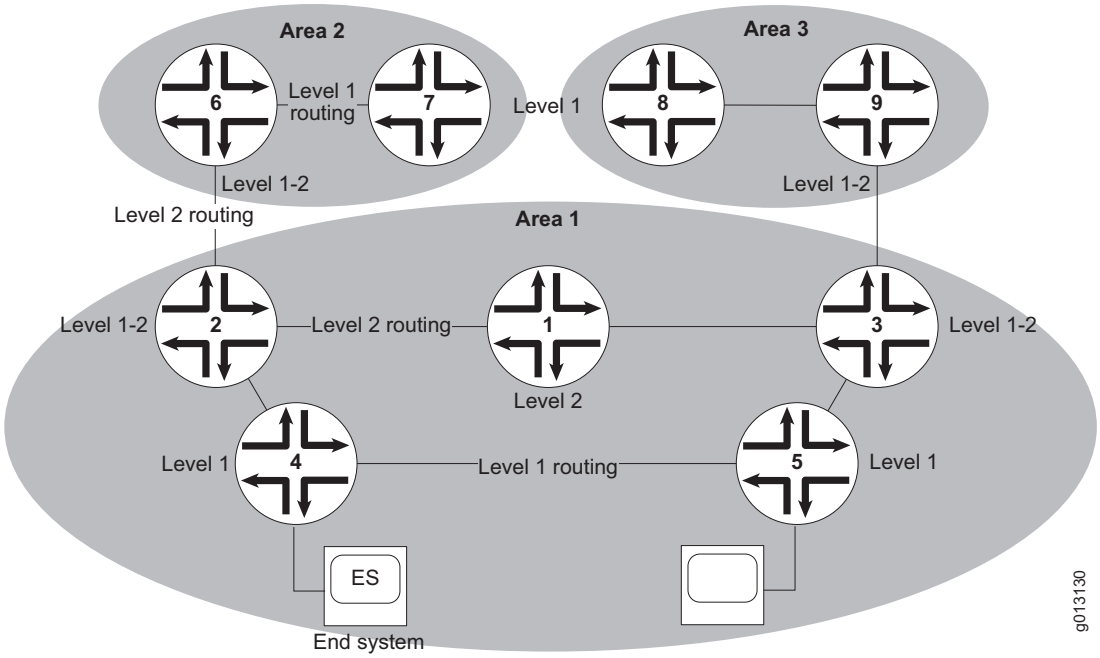
  host1(config-router-af)#**redistribute ospf level-1-2**

- Use the **no** version to disable redistribution.

Redistributing Routes Between Levels

The two-level routing hierarchy of IS-IS can lead to suboptimal path selection in certain situations. Because a level 1 router by default has knowledge only of level 1 routes, traffic from a level 1 router to a router in another area passes through the nearest level 1-2 router as its next hop. Consider the topology shown in Figure 20.

Figure 20: Example of Level 1 and Level 2 Routing



In this example, Router 4 in Area 1 considers Router 2 to be its next hop for interarea traffic, and Router 5 considers Router 3 to be its next hop for interarea traffic. Traffic from Router 4 to Router 8 passes through Router 2, requiring a total of five hops to the destination: Routers 2, 1, 3, 9, and 8. Similarly, five hops are required for traffic from Router 5 to Router 7.

Neither of these paths is optimal. For example, it would be shorter for traffic from Router 4 to take the four-hop path: Routers 5, 3, 9, and 8.

You can configure IS-IS to redistribute routes between the routing levels; this is sometimes known as route leaking between levels. The **redistribute isis ip** command enables you to specify a route filter (an access list) and the direction of leakage, as shown in the following example:

```
host1(config)#access-list leakList permit ip 100.0.0.0 0.255.255.255 any
host1(config)#router isis 1
host1(config-router)#redistribute isis ip level-1 into level-2 distribute-list leakList
host1(config-router)#redistribute isis ip level-2 into level-1 distribute-list leakList
```

When you issue the **redistribute isis ip** command and include the **route-map** keyword, you can use the map to set a route tag for a route redistributed from one IS-IS level to another. Make sure the route map you specify includes the **set tag** command that defines a tag value for the IS-IS routes to be redistributed. For details about configuring and using route maps, see *Route Maps* in *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

To redistribute IPv6 routes from one IS-IS level to another, use the **redistribute isis** command from within the IS-IS IPv6 address family.

redistribute isis

- Use to redistribute IS-IS IPv6 routes from level 1 to level 2 or from level 2 to level 1.
- Use the **route-map** keyword to specify the route map to be applied. You can use the route map to set a route tag for redistributed routes.
- Example

  host1(config-router-af)#**redistribute isis level-1 into level-2**

- Use the **no** version to stop redistribution of IPv6 routes between the specified levels.

redistribute isis ip

- Use to redistribute IS-IS IP routes from level 1 to level 2 or from level 2 to level 1.
- Specify one of the following:
  - Use the **distribute-list** keyword to specify the IP access list used to filter routes between levels. Issue the **access list** command to create a route filter to apply to the redistribution.
  - Use the **route-map** keyword to specify the route map to be applied. You can use the route map to set a route tag for redistributed routes.
- Example 1—Redistributes IS-IS IP routes between levels, filtered by an access list.

  host1(config-router)#**redistribute isis ip level-1 into level-2 distribute-list leakList**

- Example 2—Redistributes IS-IS IP routes between levels, filtered by a route map.

  host1(config-router)#**redistribute isis ip level-2 into level-1 route-map boston01**

- Use the **no** version to stop redistribution of IP routes between the specified levels.

Controlling Granularity of Routing Information

You can force the distribution of level 2 routing information to level 1 routers in other areas to improve the quality of the resulting routes, but at the cost of reduced scalability.

distribute-domain-wide

- Use to increase the granularity of routing information within a domain.
- Domainwide prefix distribution enables a routing domain running with both level 1 and level 2 IS-IS routers to distribute IP prefixes from level 2 to level 1 between areas.
- The major advantage for using domainwide prefix distribution is to improve the quality of the resulting routes within a domain by distributing more specific information.
- The major disadvantage of using domainwide prefix distribution is that it affects the scalability of IS-IS. When used, it increases the number of prefixes throughout the domain, causing increased memory consumption, transmission requirements, and computation requirements throughout the domain.
- A trade-off decision must be made between scalability and optimality.
- Issue this command from within the IS-IS IPv6 address family to increase the granularity of IPv6 routing information within a domain.
- Example

  host1(config-router)#**distribute-domain-wide**

- Use the **no** version to halt the distribution of routes from level 2 to level 1.

Configuring a Global Default Metric

You can use the **metric** command to specify a global default metric that applies to all active IS-IS interfaces. This command enables you to avoid configuring the desired metric on each active interface individually when you want all IS-IS interfaces to have the same metric, but a different value than the individual default of 10. The global default metric applies to both level 1 and level 2 interfaces unless you restrict it to one level.

If you have configured a nondefault metric on any IS-IS interface with the **isis metric** command, that value overrides the global default metric.

Reference bandwidth takes precedence over both individual and global default metrics. If you have configured a reference bandwidth, the **metric** command has no effect on interface metrics,

You can use the following commands to verify configuration of the global default metric:

- **show configuration**
- **show clns interface**
- **show clns protocol**
- **show isis database detail**

metric

- Use to apply the same default metric value to all active IS-IS interfaces. The command affects both IPv4 and IPv6 interfaces.
- Specify whether the command applies to level 1 or level 2 interfaces. If you do not specify a level, then the metric is applied to both level 1 and level 2 interfaces.
- Example

  host1(config-router)#**metric 50 level-1**

- Use the **no** version to remove the global default value. This restores the default value of 10 to all active IS-IS interfaces except for interfaces that have been individually configured with another metric value.

Configuring Metric Type

Extensions to IS-IS traffic engineering enable the use of bigger metrics. You can specify whether your router accepts, generates, or accepts and generates only old-style metrics, only new-style metrics, or both.

metric-style narrow

- Use to specify that the router generates and accepts only old-style TLV tuples.
- *Old-style TLVs* refers to TLVs having metrics with a narrow (six-bit) field with a value in the range 0–63. *New-style TLVs* refers to TLVs having metrics with a wider field, as provided for in current extensions to IS-IS traffic engineering.
- Use the **transition** option to *accept* old-style and new-style metrics; only old-style metrics are *generated*.
- Specify whether the command applies to level 1, level 2, or both.
- Example

  host1(config-router)#**metric-style narrow level-2**

- Use the **no** version to restore the default, which is to generate and accept only old-style TLVs with narrow (six-bit) metric fields.

metric-style transition

- Use to specify that the router generates and accepts both old-style and new-style TLV tuples.
- *Old style* refers to TLVs having metrics with a narrow (six-bit) field with a value in the range 0–63. *New style* refers to TLVs having metrics with a wider field, as provided for in current extensions to IS-IS traffic engineering.
- Specify whether the command applies to level 1, level 2, or both.
- Example

  host1(config-router)#**metric-style transition level-1**

- Issuing this command results in more resource usage than issuing the **metric-style narrow** or **metric-style wide** commands.
- Use the **no** version to restore the default, which is to generate and accept only old-style TLVs with narrow (six-bit) metric fields.

metric-style wide

- Use to specify that the router generates and accepts only new-style TLV tuples.
- *Old style* refers to TLVs having metrics with a narrow (six-bit) field with a value in the range 0–63. *New style* refers to TLVs having metrics with a wider field, as provided for in current extensions to IS-IS traffic engineering.
- Use the **transition** option to *accept* old-style and new-style metrics; only new-style metrics are *generated*.
- Specify whether the command applies to level 1, level 2, or both.
- Before you set a route tag for an IS-IS interface, you must issue the **metric-style wide** command to configure the router to generate and accept TLV type 135, which is a new-style tuple that contains the route tag.
- Example

  host1(config-router)#**metric-style wide level-1-2**

- Use the **no** version to restore the default, which is to generate and accept only old-style TLVs with narrow (six-bit) metric fields.

Setting the Administrative Distance

You can indicate the dependability of a routing information source by configuring the administrative distance for learned routes.

distance ip

- Use to configure the administrative distance for IS-IS learned routes.
- The distance indicates the dependability of a routing information source. A higher relative value indicates lower dependability. Preference is always given to the routes with smaller values.
- Select a value in the range 1–255. A value of 255 means discard the route.
- Example

  host1(config-router)#**distance ip 50**

- Use the **no** version to restore the default value, 115.

Configuring Default Routes

You can specify a default route within IS-IS routing domains. You can also suppress the installation of a default route to level 1-2 routers by level 1 routers.

Optionally, when you issue the **default-information originate** command and specify a route map, you can use the map to set a route tag for the default route. Make sure the route map you specify includes the **set tag** command, which defines a tag value for the default route within the IS-IS domain. For details about configuring and using route maps, see *Route Maps* in *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

default-information originate

- Use to generate a default route into an IS-IS routing domain.
- When you specify a route map with this command and the router has a route to 0.0.0.0 in the routing table, IS-IS originates an advertisement for 0.0.0.0 in its LSPs.
- When you do not specify a route map, the default route is advertised only in level 2 LSPs.
- If you specify a route map, you can use the map to set a route tag for the default route.
- For level 1 routing, look for the closest level 1-2 router to find the default route. The closest level 1-2 router is found by looking at the attach bit (ATT) in level 1 LSPs.
- The default value is disabled.
- Example1

  host1(config-router)#**default-information originate**

- Example 2

  host1(config-router)#**default-information originate route-map map3**

- Use the **no** version to disable the command.

suppress-default

- Use to prevent level 1 routers from automatically installing a default route to a level 1-2 router in order to reach destinations outside the area.
- Suppresses the level 1-2 router from indicating to level 1 routers that it can reach other areas. Consequently, the level 1 routers do not consider the level 1-2 router to be the nearest attached level 2 router and do not install default routes to it.
- This command is useful, for example, if you issue the distribute-domain-wide command, which causes the level 2 routes to be leaked into the level 1 area. The level 1 routers then have knowledge of the routes outside the area and will not need to rely on the nearest attached level 2 router for any unknown destination.
- Example

  host1(config-router)#**suppress-default**

- Use the **no** version to disable suppression of default routes.

Setting Router Type

You can specify whether the router behaves as an IS-IS station router, area router, or both.

is-type

- Use to configure the router to act as either a station router (level 1), an area router (level 2), or as both a station router and an area router (level-1-2).
- Always configure the type of IS-IS router.
- Level-1-2 is the default.
- Example

  host1(config-router)#**is-type level-2-only**

- Use the **no** version to restore the default value, level-1-2.

Summarizing Routes

You can summarize routes redistributed into IS-IS or within IS-IS by creating aggregate addresses for the routes. Use the **summary-address** command for IP routes and the **summary-prefix** command for IPv6 routes.

Optionally, you can set a route tag for an IS-IS aggregate (summary) address by including the **tag** keyword and a numeric tag value in the command.

summary-address
summary-prefix

- Use to create aggregate addresses of routes that are redistributed from other protocols in the routing table or distributed between level 1 and level 2 by a summary address. This process is called *route summarization*.
- A single summary address includes groups of addresses for a given level.
- Use the **summary-address** command for IP routes. Use the **summary-prefix** command for IPv6 routes.
- The metric value is used when the router advertises the summary address. When the metric value is not used, the value of the lowest cost route (the default) is used.
- This command reduces the size of the neighbor's routing table and improves stability because a summary advertisement depends on many more specific routes.
- A disadvantage of summary addresses is that other routes might have less information to calculate the optimal routing table for all individual destinations.
- Use the optional **tag** keyword to specify a tag value for an IS-IS summary address. The tag value must be a number in the range 1–4294967295.
- Example 1—For IP routes

  host1(config-router)#**summary-address 10.2.0.82 255.255.0.0 level-1-2 tag 34**

- Example 2—For IPv6 routes

  host1(config-router-af)#**summary-prefix 2001:2000::0/8 level-1 metric 10 tag 100**

- Use the **no** version to restore the default, the value of the lowest-cost route.

Avoiding Transient Black Holes

When you start or reload a transit router that is running both IS-IS and BGP, the router is temporarily unavailable to the routing domain. Other routers in that routing domain must select alternative paths to destinations that used the transit router. When the transit router becomes available again, the other routers soon select it again as the optimal path to those destinations.

The other routers select the transit router again before it has loaded the complete BGP routing table. Because the transit router does not yet have all the reachability information that is needed to reach some external destinations, traffic to destinations that were not learned by means of the IGP is dropped until the transit router has complete external reachability information again. This condition is known as a *transient black hole*.

You can use the overload bit to avoid these black holes. When the overload bit is set in the LSP header, other routers in the domain do not include the transit router in their SPF calculations and thus do not use that router for traffic forwarding.

When the transit router boots, it begins establishing adjacencies with its neighbors. As soon as it establishes an adjacency, it creates (or updates) its LSP, sets the overload bit in the LSP header, and transmits the LSP with the current neighbor information. By sending the updated LSP with the overload bit set immediately after forming the first adjacency, IS-IS reduces the convergence time across the network.

If IS-IS waits for all adjacencies to be up before it sends the updated LSP with the overload bit set, the other routers in the domain still have the transit router's old LSP and continue to forward transit traffic to the transit router until all adjacencies are formed. That traffic is lost.

Waiting for BGP Convergence

When BGP converges, the transit router again has the reachability information it needs to forward traffic to destinations that are not directly connected. Typically, you then want the transit router to clear the overload bit in its LSP and retransmit the LSP to inform the other routers in the domain that they can use it as a transit router.

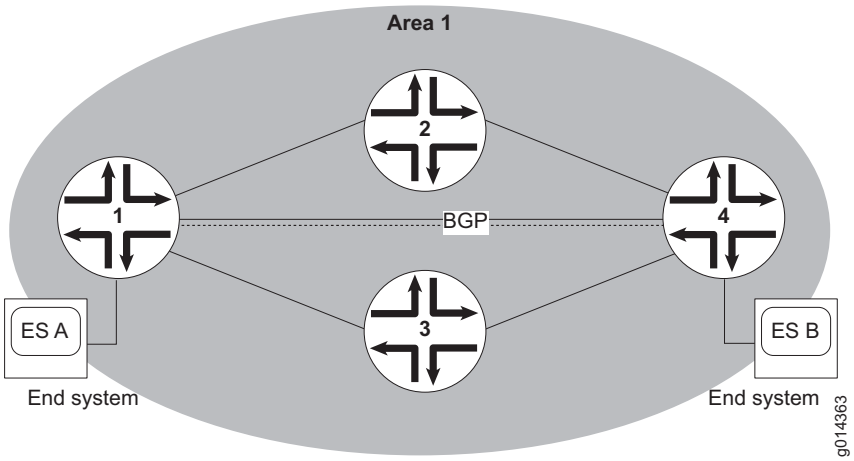
BGP is assumed to have converged when all of the following conditions have been met:

- 90 percent of BGP peers have reached an established state,
- The transit router has received an end-of-rib marker from all IBGP peers that advertise the graceful-restart capability.
- The average rate of learning new routes has dropped to a low level.

Example Topology

Figure 21 shows a sample topology where source end system A is communicating with destination end system B through routers 1, 2, 3, and 4.

Figure 21: Transit Router Topology



The transit routers, 2 and 3, learn the route to B from BGP. In a steady state environment, the BGP routing tables are synchronized on all the transit routers.

Suppose the traffic forwarding path is currently A –> 1 –> 2 –> 4 –> B. If transit router 2 goes down, the network converges to the alternative path, A –> 1 –> 3 –> 4 –> B. Because transit router 3 already had synchronized its BGP routing tables, traffic forwarding continues without delay.

When transit router 2 reloads, it establishes adjacencies with routers 1 and 4, and sends out its LSP advertising its neighbors. While router 2 begins to synchronize its BGP routes, the network reconverges to the original path of A –> 1 –> 2 –> 4 –> B. Traffic from A to B is forwarded to router 2. Typically, BGP has not converged by then, so router 2 does not have the BGP route that it needs to forward the traffic, and drops the packets, resulting in a black hole until the BGP convergence is complete.

You can avoid this black hole by configuring the overload bit for the transit router. In this circumstance, router 2 sends out its LSP with the overload bit set in its header as soon as it reloads, before it establishes all adjacencies. The bit set in the header indicates to all the routers in the domain that router 2 is overloaded and not to use it to carry transit traffic. The forwarding path continues to be the alternative path, A –> 1 –> 3 –> 4 –> B, even after router 2 reloads.

When BGP convergence is complete at router 2, router 2 sends out a new LSP with the overload bit cleared. The other routers then include router 2 in their SPF calculations and revert to the original path, of A –> 1 –> 2 –> 4 –> B.

Suppression for IS-IS Graceful Restart

When graceful restart is configured on the transit router, the black hole avoidance feature is suppressed.

Configuration

You can configure the transit router to set the overload bit when it reloads and to then wait for a specified interval before it clears the bit and retransmits its LSP. More commonly, and to avoid the transient black holes, you configure the transit router to wait for BGP to converge, and specify an interval it waits after convergence before it clears the bit and retransmits its LSP.

set-overload-bit

- Use to configure the router to set the overload bit in the header of its nonpseudonode LSPs.
- While the overload bit is set, other routers in the domain do not include this router in their shortest-path-first (SPF) calculations. Consequently, the other routers do not detect any paths through this router and do not forward traffic through this router. However, IP prefixes directly connected to this router are still reachable. When the bit is cleared, the router is again included in SPF calculations.
- You can set the overload bit for a number of reasons, including the following:
  - To prevent traffic through the router from disappearing into transient black holes.
  - To reduce routing table inaccuracies caused by router problems such as memory shortage.
  - To prevent real traffic from flowing through a router to an IS-IS network, such as might be the case for a test router connected to a production network.
- Use the **on-startup** keyword to set the overload bit when the router reboots and to specify a period in seconds that IS-IS waits after the reboot before it clears the overload bit.
- Use the **on-startup wait-for-bgp** keywords to instruct IS-IS to set the overload bit when the router reboots and then wait until BGP has completed convergence after the reload before IS-IS clears the overload bit. You can specify a maximum interval that IS-IS waits for BGP notification. When that interval passes, IS-IS clears the overload bit. If you do not specify an interval, IS-IS waits a default 600 seconds and then clears the overload bit.
- If you issue the **on-startup** keyword but do not issue the **wait-for-bgp** keyword, then you must specify the number of seconds that IS-IS waits after a reload before clearing the overload bit.
- If you issue both the **on-startup** keyword and the **wait-for-bgp** keyword, you cannot specify a time interval for **on-startup** but can optionally do so for **wait-for-bgp**.
- By default, the overload bit is not set.

- Example 1

  host1(config-router)#**set-overload-bit**

- Example 2

  host1(config-router)#**set-overload-bit on-startup 900**

- Example 3

  host1(config-router)#**set-overload-bit on-startup wait-for-bgp 450**

- Use the **no** version to disable the setting.

Ignoring LSP Errors

You can configure the router to ignore rather than purge LSPs received with errors.

ignore-lsp-errors

- Use to enable your router to ignore rather than purge IS-IS LSPs that are received with internal checksum errors.
- Under normal conditions, the IS-IS protocol definition requires that received LSPs with incorrect data link checksums are to be purged by the receiver. This causes the LSP initiator to regenerate LSPs. If a network link causes data corruption while still delivering LSPs with correct data link checksums, a continuous cycle of regenerating and purging LSPs may result. This can render the network nonfunctional. Enabling this command prevents this continuous cycle from occurring because LSPs are ignored rather than purged.
- Example

  host1(config-router)#**ignore-lsp-errors**

- Use the **no** version to disable the function.

Logging Adjacency State Changes

You can configure the router to log messages that track when adjacencies change state between up and down.

log-adjacency-changes

- Use to generate log messages that track IS-IS adjacency state changes (up or down).
- The default is not to log adjacency state changes.
- Recommended for monitoring large networks.
- The system logs messages by using the router error message facility.
- Specify the minimum severity (0–7) or verbosity (low, medium, high) of this log category's messages.

- You can also use the **system log** command to generate the desired log messages.
- Example

  host1(config-router)#**log-adjacency-changes severity 3 verbosity low**

- Use the **no** version to disable the function.

Configuring LSP Parameters

You can specify the following parameters for LSPs:

- Maximum transmission unit (MTU)
- Transmission rate
- Generation rate
- Maximum lifetime

lsp-gen-interval

- Use to set the minimum interval rate that LSPs are generated on a per-LSP basis.
- You can set an interval value in the range 0–120 seconds.
- The default interval value is 5 seconds. When a link is changing state at a high rate, the default value limits the signaling of the changing state to once every 5 seconds. Because the generation of an LSP may cause all routers in the area to perform the SPF calculation, controlling this interval can have an areawide effect.
- When you raise this interval, you reduce the load on the network imposed by a rapidly changing link.
- Example

  host1(config-router)#**lsp-gen-interval level-2 30**

- Use the **no** version to restore the default value, 5.

lsp-mtu

- Use to specify the MTU LSP size in bytes. The size must be less than or equal to the smallest MTU of any link in the area.
- Use this command to limit the size of LSPs generated by this router only. The router can receive LSPs of any size up to the maximum.
- You can set the value in the range 128–9180.
- The default LSP MTU value is 1497.
- When a very large amount of information is generated by a single router, we recommend that you increase the LSP MTU. However, the default MTU is usually sufficient.

- If the MTU of a link is lowered to less than 1500 bytes, the LSP MTU must be lowered accordingly on each router in the network. If this is not done, routing may become unpredictable.
- Example

  host1(config-router)#**lsp-mtu 1500**

- Use the **no** version to restore the default value, 1497.

lsp-refresh-interval

- Use to set the LSP rate at which locally generated LSPs are periodically transmitted.
- The refresh interval determines the rate at which the router software periodically transmits the route topology information that it originates. These transmissions refresh the link-state information, reaffirming that the router is still up and that the link-state information in the LSP is still valid.
- You can set the interval rate in the range 1–65535 seconds; the default is 900 seconds.
- LSPs must be periodically refreshed before their lifetimes expire. The refresh interval must be less than the LSP lifetime specified by **max-lsp-lifetime**.
- In the unlikely event that link stage database corruption is undetected, reducing the refresh interval reduces the amount of time that the corruption can persist.
- Increasing the interval reduces the link utilization caused by the flooding of refreshed packets.
- Example

  host1(config-router)#**lsp-refresh-interval 1000**

- Use the **no** version to restore the default value, 900 seconds.

max-lsp-lifetime

- Use to set the maximum time that LSPs persist without being refreshed.
- You can select a maximum time in the range 1–65535 seconds.
- The default value is 1200 seconds (20 minutes).
- You might need to adjust the maximum LSP lifetime if you change the LSP refresh interval with the **lsp-refresh-interval** command. The maximum LSP lifetime must be greater than the LSP refresh interval.
- Example

  host1(config-router)#**max-lsp-lifetime 1500**

- Use the **no** version to restore the default value, 1200 seconds.

Specifying the SPF Interval

You can configure how often the router performs the shortest-path-first (SPF) calculation. IS-IS runs SPF calculations in response to any change in its link-state database. Because SPF calculation is processor intensive, increasing the SPF interval reduces the processor load of the router, but can slow down the rate of convergence.

Topology changes in a network cause all routers involved in the change to regenerate their LSDB and flood new LSPs throughout the network. Therefore, a router that receives a new LSP is likely to receive more LSPs in the following seconds. An immediate response to a given change is going to miss the subsequent topology changes and spend CPU time. When many changes are taking place, a slower response to each change makes more sense.

IS-IS enables the router to respond quickly to an isolated network event, but to slow the response exponentially when many triggering events are taking place in rapid succession. SPF calculations are performed at exponentially increasing intervals until the maximum interval set by the **spf-interval** command is reached.

The first SPF calculation is performed immediately when the LDSB changes. If another calculation-triggering event occurs, the router waits 1 second before performing the SPF calculation. If another event occurs, the router waits 2 seconds before performing the SPF calculation. The interval between a triggering event and the corresponding SPF calculation continues to increase exponentially: 4 seconds, 8 seconds, 16 seconds, and so on. When the maximum configured interval is reached, the interval reverts back to immediate response mode for the next triggering event.

If no calculation-triggering network events have occurred by the end of any given back-off interval, the router reverts back to immediate response mode.

spf-interval

- Use to set the maximum interval between SPF calculations.
- You can select an interval value in the range 0–120 seconds.
- The default value is 5 seconds.
- If you do not specify **level-1** or **level-2**, the interval applies to both level 1 and level 2.
- SPF calculations are performed only when the topology of the area changes. They are not performed when external routes change.
- Example

  host1(config-router)#**spf-interval level-2 30**

- Use the **no** version to restore the default value, 5 seconds.

Defining the SPF Route Calculation Level

The IS-IS protocol uses the Dijkstra algorithm to compute IP node metrics when a change occurs within the IS-IS network. This calculation results in the IS-IS router containing a shortest-path tree (SPT) that maps the shortest path to each node in the IS-IS network.

By default, the router uses a partial route calculation (PRC) SPF to determine the next hop (when required). This partial computation occurs when the router receives link-state PDUs (LSPs) with only changes relating to IP prefixes (for example, the addition of a new IP prefix, change in attributes of an existing IP prefix, or the removal of an existing IP prefix).

Because changes in IP prefixes happen more frequently than other events, using the PRC SPF results in faster IS-IS convergence and saves router resources. However, you can also specify that the router always use full SPF, recalculating the entire SPT, when resolving any IS-IS state changes.

full-spf-always

- Use to enable and disable full SPF calculations for IS-IS network changes.
- Example

  host1(config-router)#**full-spf-always**

- Use the **no** version to restore partial route calculation (PRC) mode for SPF calculations.

Setting CLNS Parameters

You can specify transmission rates for ES and IS hello packets, the period for which the router considers ES and IS hello packets to be valid, and name-to-network service access point mappings.

clns configuration-time

- Use to specify the rate (in seconds) at which ES hello and IS hello packets are sent.
- The hello packet recipient creates an adjacency entry for the router that sent it. If the next hello packet is not received within the specified interval, the adjacency times out, and the adjacent node is determined to be unreachable.
- In most cases, leave these parameters at their default value, which is 10 seconds.
- Example

  host1(config)#**clns configuration-time 240**

- Use the **no** version to restore the default value, 10 seconds.

clns holding-time

- Use to enable sender of an ES hello or IS hello packet to specify the length of time you consider the information in these packets to be valid.
- In most cases, leave these parameters at their default value, which is 30 seconds.
- Example

  host1(config)#**clns holding-time 900**

- Use the **no** version to restore the default value, 30 seconds.

clns host

- Use to define a name-to-NSAP mapping that can then be used with commands requiring NSAPs.
- The default is that no mapping is defined.
- The assigned NSAP name is displayed, where applicable, in **show** commands.
- The first character can be either a letter or a number.
- This command is generated after all other CLNS commands when the configuration file is parsed. As a result, the NVRAM version of the configuration cannot be edited to specifically change the address defined in the original **clns host** command. You must specifically change any commands that refer to the original address. This affects commands that accept names, such as the **net** command.
- Enables dynamic resolution of hostnames to system IDs (within the NSAP address). The hostname mapping is sent in the LSPs within the Dynamic Hostname type-length-value (TLV type 137). Display the TLV by issuing the **show isis database detail** command.
- Use the **show hosts** command to display the mapping.
- Example

  host1(config)#**clns host**

- Use the **no** version to restore the default state of no mapping defined.

Setting the Maximum Parallel Routes

You can configure how many parallel routes IS-IS supports to a destination.

maximum-paths

- Use to control the maximum number of parallel routes IS-IS can support.
- You can select a number of routes (or paths) in the range 1–16.
- The default number for IS-IS is 4 paths.
- Example

  host1(config-router)#**maximum-paths 12**

- Use the **no** version to restore the default value, 4.

Configuring a Virtual Multiaccess Network

You can specify that interfaces within a given mesh group act as a virtual multiaccess network.

isis mesh-group

- Use when you want interfaces in the same mesh group to act as a virtual multiaccess network.
- LSPs seen on one interface in a mesh group are not flooded to another interface in the same mesh group.
- Example

  host1(config-if)#**isis mesh-group blocked**

- Use the **no** version to disable the feature.

Configuring Table Maps

You can use the **table-map** command to apply a specified route map as a policy filter on an IS-IS route before the route is installed in the routing table. The route map you apply must contain one or more **set** commands to modify route attributes.

table-map

- Use to apply a policy to modify distance, level, metric, metric type, origin, preference, route type, or tag values of IS-IS routes about to be added to the IP routing table.
- The router applies the new route map to all routes currently in the forwarding table and those about to be installed in the forwarding table.
- If any previously redistributed routes are changed as a result of applying the route map, the router redistributes these routes again with the changes caused by the route map.
- The router removes from the forwarding table any old routes that are now disallowed by the specified route map.
- Issue the command from the IS-IS IPv6 address family to apply a specified route map as a policy filter on an IS-IS IPv6 route before the route is installed in the routing table. IS-IS IPv6 supports only a single table map.
- Example

  The following commands apply a policy (route map) named metricTypeExt to modify the metric type of IS-IS routes configured with a route tag value of 33.

  host1(config)#**route-map metricTypeExt permit 5**
  host1(config-route-map)#**match tag 33**
  host1(config-route-map)#**set metric-type external**
  host1(config-route-map)#**exit**
  host1(config)#**router isis marketing**
  host1(config-router)#**table-map metricTypeExt**
  host1(config-router)#**exit**
  host1(config)#**exit**

- Use the **no** version to halt application of the route map.

Configuring Graceful Restart

To enable IS-IS graceful restart (also known as nonstop forwarding, or NSF) on the router, you must first issue the **nsf ietf** command (in Router Configuration mode). You can then configure one or more optional timing parameters for graceful restart on the router.

To enable IS-IS graceful restart and configure optional graceful restart parameters:

1. Specify a previously configured IS-IS routing process to access Router Configuration mode. (For information about enabling IS-IS on the router, see *Enabling IS-IS for IP Routing* on page 326.)

   host1(config)#**router isis engineering**
   host1(config-router)#

2. Enable the IS-IS graceful restart mechanism for the router.

   host1(config-router)#**nsf ietf**

3. (Optional) Configure one or more of the following timing parameters for the restarting router:

   - Set the maximum time in seconds that the router waits before completing the restart process.

     host1(config-router)#**nsf interface wait 30**

   - Set the time interval in seconds between restart requests sent by the router.

     host1(config-router)#**nsf t1 interval 60**

   - Set the number of times that the router resends unacknowledged restart requests.

     host1(config-router)#**nsf t1 retry-times 3**

   - Set the maximum time in seconds that the router waits for the LSP database to synchronize. You must configure this parameter separately for each IS-IS level at which the router operates.

     host1(config-router)#**nsf t2 level-1 70**
     host1(config-router)#**nsf t2 level-2 50**

   - Set the maximum time in seconds that the restarting router waits before setting the overload bit to indicate that the graceful restart operation has failed. You can use either of the following methods:

     - Set the wait time manually to the specified number of seconds.

       host1(config-router)#**nsf t3 manual 80**

     - Specify that router obtain the wait time from neighboring IS-IS routers to which it has active adjacencies.

       host1(config-router)#**nsf t3 adjacency**

4. (Optional) Issue the **show isis nsf** command from Privileged Exec mode to verify the graceful restart configuration.

   ```
   host1(config-router)#exit
   host1(config)#exit
   host1#show isis nsf
   ```

   For more information about monitoring graceful restart, see the **show isis nsf** command description in *Monitoring IS-IS Parameters* on page 373 and the **show clns neighbors detail** command description in *Displaying CLNS* on page 385.

nsf ietf

- Use to enable the IS-IS graceful restart mechanism on the router.
- Graceful restart, which is also known as nonstop forwarding (NSF), allows an IS-IS router to restart with minimal routing disruption to the network.
- Example

  ```
  host1(config-router)#nsf ietf
  ```

- Use the **no** version to restore the default state for IS-IS graceful restart on the router, disabled.

nsf interface wait

- Use to specify the maximum amount of time, in seconds, that an IS-IS process on a restarting router waits for all interfaces with IS-IS adjacencies to come up before completing the restart process.
- You can specify a value in the range 5–120 seconds.
- Example

  ```
  host1(config-router)#nsf interface wait 45
  ```

- Use the **no** version to restore the default maximum wait time, 10 seconds.

nsf t1

- Use to specify either the interval between IS-IS restart requests sent by the router or the number of times that the router resends unacknowledged restart requests.
- Use the **interval** keyword to specify the number of seconds, in the range 5–120, between restart requests sent by the router on a particular IS-IS interface to neighboring IS-IS routers in the network.
- Use the **retry-times** keyword to specify the number of times, in the range 1–3, that the router tries to resend unacknowledged restart requests.
- The restarting router stops sending restart requests after it receives an acknowledgment.

- Example 1

  host1(config-router)#**nsf t1 interval 90**

- Example 2

  host1(config-router)#**nsf t1 retry-times 2**

- Use the **no** version to restore the default time interval, 5 seconds, or the default number of retry attempts, 1.

nsf t2

- Use to specify the maximum amount of time, in seconds, that a restarting router waits for the LSP database to synchronize.
- You must configure independent instances of the T2 timer for each IS-IS level at which the router operates. This requirement means that for a level 1-2 router, you must issue this command twice: first to configure the timer for level 1, and a second time to configure it for level 2.
- Use either the **level-1** keyword to set the T2 wait time for level 1 routing, or the **level-2** keyword to set the wait time for level 2 routing.
- You can specify a value in the range 5–120 seconds for each level.
- Example—Configures the T2 wait time for a level 1-2 IS-IS router

  host1(config-router)#**nsf t2 level-1 70**
  host1(config-router)#**nsf t2 level-2 50**

- Use the **no** version to restore the default T2 wait time, 30 seconds.

nsf t3

- Use to specify the maximum amount of time, in seconds, that the restarting router waits before setting the overload bit.
- The restarting router sets the overload bit to indicate that the LSP database has not been synchronized and the IS-IS graceful restart operation has failed.
- You must use one of the following methods to set the T3 wait time:
  - Use the **manual** keyword and a value in the range 5–120 seconds to set the T3 wait time manually.
  - Use the **adjacency** keyword to specify that the restarting router should obtain its T3 wait time from neighboring IS-IS routers that have active adjacencies to this router. This option sets the wait time to the minimum of the remaining times specified in the restart TLVs contained in the hello packets that the router receives from its neighbors.
- Example1

  host1(config-router)#**nsf t3 manual 120**

- Example 2

  host1(config-router)#**nsf t3 adjacency**

- Use the **no** version to restore the default T3 wait time, 30 seconds.

Summary Example

```
host1(config)#router isis floor12
host1(config-router)#net 47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00
host1(config-router)#exit
host1(config)#interface atm 0/1
host1(config-if)#ip router isis floor12 tag 24
host1(config-if)#isis mesh-group blocked
host1(config-if)#exit
host1(config)#interface atm 1/0
host1(config-if)#ip router isis floor12
host1(config-router)#distribute-domain-wide
host1(config-router)#distance 100 ip
host1(config-router)#default-information originate route-map 9
host1(config-router)#is-type level-1-2
host1(config-router)#summary-address 10.2.0.82 255.255.0.0 level-1-2 tag 90
host1(config-router)#set-overload-bit on-startup wait-for-bgp 450
host1(config-router)#ignore-lsp-errors
host1(config-router)#log-adjacency-changes
host1(config-router)#lsp-mtu 1500
host1(config-router)#lsp-refresh-interval 1000
host1(config-router)#lsp-gen-interval level-2 30
host1(config-router)#max-lsp-lifetime 1500
host1(config-router)#spf-interval level-2 30
host1(config-router)#maximum-paths 16
host1(config-router)#redistribute static ip route-map 5
host1(config-router)#nsf ietf
host1(config-router)#nsf t2 level-1 70
host1(config-router)#nsf t2 level-2 50
host1(config-router)#nsf t3 adjacency
host1(config-router)#exit
host1(config)#clns configuration-time 120
host1(config)#clns holding-time 600
```

Configuring IS-IS for MPLS

IS-IS has several commands to provide support for MPLS. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS*, for a detailed discussion of MPLS. If you configure your tunnel with the **tunnel mpls autoroute announce isis** command, MPLS attempts to register the tunnel endpoint with IS-IS. You must enable this registration with IS-IS by issuing the **mpls traffic-eng** command.

When you configure a node as the downstream endpoint of an LSP, you must provide a stable interface as the router ID for the endpoint. Typically you select a loopback interface because of its inherent stability. Use the **mpls traffic-eng router-id** command to specify the router ID.

By default, IS-IS always uses the MPLS tunnel to reach the MPLS endpoint. Best paths determined by IS-IS SPF calculations are not considered. You can enable the consideration of best paths by issuing the **mpls spf-use-any-best-path** command. As a result, IS-IS considers metrics for IGP paths and the tunnel metric, and might forward traffic along a best path, through the MPLS tunnel, or both.

Several **show** commands enable monitoring of MPLS information. See *Monitoring IS-IS* on page 372 for more information.

MPLS traffic engineering requires that IS-IS generate the new-style TLVs that enable wider metrics. Use the **metric-style wide** command to generate the new-style TLVs. If you are using some IS-IS routers that still do not understand the new-style TLVs, use the **metric-style transition** command. See *Extensions for Traffic Engineering* on page 317 and *Configuring Global IS-IS Parameters* on page 342 for detailed information about using the **metric-style** commands.

mpls spf-use-any-best-path

- Use to enable SPF calculations to consider the IGP (IS-IS) best paths as well as the MPLS tunnel for forwarding traffic to the MPLS endpoint.
- By default, the MPLS tunnel is always selected for traffic to the tunnel endpoint; IGP paths are not considered. For traffic beyond the endpoint, the tunnel is considered equally with any other path.
- Example

  host1(config-router)#**mpls spf-use-any-best-path**

- Use the **no** version to disable the use of IGP best paths.

mpls traffic-eng

- Use to enable flooding of MPLS traffic engineering link information into the specified IS-IS level. Flooding is disabled by default.
- Example

  host1(config-router)#**mpls traffic-eng level-1**

- Use the **no** version to disable flooding.

mpls traffic-eng router-id

- Use to specify a very stable interface to be used as a router ID for MPLS traffic engineering. Typically you specify a loopback interface to provide the greatest stability, because this is flooded to all nodes. The interface acts as the destination node for tunnels originating at other nodes.
- Example

  host1(config-router)#**mpls traffic-eng router-id loopback 0**

- Use the **no** version to remove the interface as a router ID.

Using IS-IS Routes for Multicast RPF Checks

You can use the **ip route-type** command to specify whether IS-IS routes are available for only unicast forwarding protocols or only multicast reverse-path forwarding (RPF) checks. Routes available for unicast forwarding appear in the unicast view of the routing table, whereas routes available for multicast RPF checks appear in the multicast view of the routing table.

ip route-type

- Use to specify whether IS-IS routes are available only for unicast forwarding, only for multicast reverse-path forwarding checks, or for both.
- Use the **show ip route** command to view the routes available for unicast forwarding.
- Use the **show ip rpf-routes** command to view the routes available for multicast reverse path forwarding checks.
- By default, IS-IS routes are available for both unicast forwarding and multicast reverse path forwarding checks.
- Example

  host1(config)#**router isis**
  host1(config-router)#**ip route-type unicast**

- Use the **no** version to restore the default value, both.

Configuring the BFD Protocol for IS-IS

The **isis bfd-liveness-detection** command configures the Bidirectional Forwarding Detection (BFD) protocol for IS-IS. The BFD protocol uses control packets and shorter detection time limits to more rapidly detect failures in a network. Also, because they are adjustable, you can modify the BDF timers for more or less aggressive failure detection.

When you issue the **isis bfd-liveness-detection** command on an IS-IS peer, the peer establishes BFD liveness detection with all BFD-enabled IS-IS peers. When the local peer receives an update from a remote IS-IS peer—if BFD is enabled and if the session is not already present—the local peer attempts to create a BFD session to the remote peer.

Each adjacent pair of peers negotiates an acceptable transmit interval for BFD packets. The negotiated value can be different on each peer. Each peer then calculates a BFD liveness detection interval. When a peer does not receive a BFD packet within the detection interval, it declares the BFD session to be down and purges all routes learned from the remote peer.

NOTE: Before the router can use the **isis bfd-liveness-detection** command, you must specify a BFD license key. To view an already configured license, use the **show license bfd** command.

For general information about configuring and monitoring the BFD protocol, see *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.

isis bfd-liveness-detection

- Use to enable BFD (bidirectional forwarding detection) and define BFD values to more quickly detect IS-IS data path failures.
- The peers in an IS-IS adjacency use the configured values to negotiate the actual transmit intervals for BFD packets.
  - You can use the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-interval** keyword to specify the same value for both of those intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value. The default value is 300 milliseconds.
- You can use the **multiplier** keyword to specify the detection multiplier value. The calculated BFD liveness detection interval can be different on each peer. The multiplier value is roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. The default value is 3.
- For details on liveness detection negotiation, see *Negotiation of the BFD Liveness Detection Interval* section in *JUNOSe IP Services Configuration Guide, Chapter 5, Configuring BFD*.
- You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.
- Example

  host1(config)#**isis bfd-liveness-detection minimum-interval 800**

- Use the **no** version to disable BFD on the IS-IS interface.

Disabling the IS-IS Protocol

The **protocol shutdown** command disables the IS-IS protocol but does not remove any IS-IS configuration. In addition, even though the router does not participate in IS-IS routing after you issue the **protocol shutdown** command, you can continue to configure IS-IS.

Issuing the **protocol shutdown** command:

- Clears the LSP database
- Removes all IS-IS routes in the routing information database (RIB)
- Deletes all adjacencies with the IS-IS instance

NOTE: Rebooting the router does not affect the state of the IS-IS protocol.

protocol shutdown

- Use to disable the IS-IS protocol without removing the IS-IS configuration.
- Example

  host1(config-router)#**protocol shutdown**

- Use the **no** version to reenable the IS-IS protocol.

Monitoring IS-IS

The CLI has commands available for monitoring IS-IS parameters and CLNS parameters.

System Event Logs

To troubleshoot and monitor IP, use the following system event logs:

- isisAdjChange—IS-IS adjacency up or down events
- isisAdjPackets—IS-IS adjacency hello packets
- isisBfdEvents—IS-IS interactions with BFD
- isisChecksumErr—IS-IS checksum errors
- isisGeneral—IS-IS system notifications
- isisHelloGeneral—IS-IS system notifications
- isisHelloPackets—IS-IS hello packets
- isisipv6Log—IS-IS IPv6 notifications
- isisLdpEvents—IS-IS interactions with LDP
- isisLocalUpdate—IS-IS local LSP packets
- isisMplsTeAdvertisements—IS-IS MPLS traffic engineering advertisements
- isisMplsTeEvents—IS-IS MPLS traffic engineering
- isisNsfEvents—IS-IS nonstop forwarding events during warm starts
- isisProtocolErr—IS-IS protocol errors
- isisSnpPackets—IS-IS complete sequence numbers PDU (CSNP) and partial sequence numbers PDU (PSNP) packets
- isisSpfEvents—IS-IS Shortest Path First (SPF)

- isisSpfStatistics—IS-IS SPF timing and statistic data
- isisSpfTriggers—IS-IS SPF triggering
- isisUpdate Packets—IS-IS LSP packets sent or received

For more information about using event logs, see the *JUNOSe System Event Logging Reference Guide, Chapter 1,* .

Monitoring IS-IS Parameters

You can monitor the IS-IS link-state database and IS-IS debug information. Use the commands in this section to:

- Display router information.
- Display information about SPF calculations.
- Monitor IS-IS summary address information.
- Display debug information.
- Display host.
- Display information about MPLS tunnels.
- Clear adjacencies.
- Display paths to intermediate systems.
- Display information about the settings for IS-IS graceful restart.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

clear isis adjacency

- Use to remove entries from the adjacency database.
- Specify a hostname or the system ID of a neighbor to clear only adjacencies with that neighbor.
- Specify no options to remove all adjacencies from the database.
- Example

  host1#**clear isis adjacency**

- There is no **no** version.

debug isis

- Use to obtain debug-related information about certain parameters.
- This command manipulates the same log as the Global Configuration **log** commands.
- You can select from these parameters:
  - **adj-packets**—IS-IS adjacency-related packets
  - **mpls traffic-eng advertisements**—MPLS traffic-engineering agent advertisements
  - **mpls traffic-eng agents**—MPLS traffic-engineering agents
  - **snp-packets**—IS-IS CSNP/PSNP packets
  - **spf-events** —IS-IS Shortest Path First events
  - **spf-statistics**—IS-IS SPF timing and statistic data
  - **spf-triggers**—IS-IS SPF triggering events
  - **update-packets**—IS-IS update-related packets
- Example

  host1#**debug isis adj-packets**

- Use the **no** version to disable debugging display.

show hosts

- Use to display the name-to-NSAP mappings defined with the **clns host** command.
- Field descriptions
  - Static Host Table
    - name—Name assigned to the host
    - ip address—Host IP address
    - type—Type of host
    - username—Username necessary to access the host
    - password—Password necessary to access the host
  - Clns Host Alias Table
    - name—Name of the host alias
    - area address—Area address of the host alias
    - system ID—Six-byte value of the host alias
    - type—Type of host alias
- Example

```
host1:abc#show hosts
Static Host Table
-----------------
name   ip address   type   username    password
----   ----------   ----   ---------   --------
jkk    10.10.0.73   ftp    anonymous   null
```

```
Clns Host Alias Table
---------------------
name  area address                        system ID         type
----- ------------------------------ ----------------- ------
fred  47.0005.80FF.F800.0000.0001.0001 0000.0000.0011.00 static
karen 47.0005.80FF.F800.0000.0001.0001 0000.0000.0012.00 static
```

show isis database

- Use to display IS-IS link-state database information.
- Request specific **show isis database** statistics by selecting from these options:
  - *lspid*—Link-state protocol ID in form xxxx.xxxx.xxxx.yy-zz
  - *hostname*—Link-state database information for the specified hostname
  - **detail**—Detailed link-state database information; if this option is not specified, a summary display is provided
  - **l1**—Level 1 routing link-state database
  - **l2** —Level 2 routing link-state database
  - **level-1**—Level 1 routing link-state database
  - **level-2**—Level 2 routing link-state database
- Each option can be entered in an arbitrary string within a single command entry.
- Field descriptions
  - LSPID—LSP identifier
  - LSP Seq Num—Sequence number for the LSP. Enables other routers to determine if they have received the latest information from source.
  - LSP Checksum—Checksum of the LSP packet
  - LSP Holdtime—Number of seconds that the LSP remains valid
  - ATT—Attach bit; indicates that the router is a level 2 router and can reach other areas
  - P—P bit; detects whether intermediate system is capable of area partition repair
  - OL—Overload bit; determines whether intermediate system is congested
  - Area Address—Area addresses that can be reached from the router
  - NLPID—ISO network layer protocol identifier
  - IP Address—IP address of the interface
  - Hostname—Hostname of the router
  - Router ID—ID configured on the router
  - Metric —Metric that indicates either of the following costs:
    - Cost of adjacency between the originating router and the advertised neighbor
    - Cost between the advertising router and the advertised destination

- IPv4 Interface Address—Address of the interface
- IPv4 Neighbor Address—Address of a neighbor
- Maximum link bandwidth—Bandwidth capacity of the link in bits per second
- Reservable link bandwidth—Amount of bandwidth reservable on the link (whether reserved or not)
- Unreserved bandwidth—Amount of bandwidth available for reservation on the link
- TE default metric—Traffic engineering default metric value
- Tag value(s)—Route tag assigned to the IS-IS interface, if configured

- Example 1

```
host1#show isis database
IS-IS Level-1 Link State Database
LSPID LSP    Seq Num     LSP Checksum LSP Holdtime ATT/P/OL
0000.0000.004E.00-00  0x000013F5   0x8BAA 1198         0/0/0
0000.0000.3333.00-00* 0x0000020F   0xEA1E 1199         0/0/0
0000.0000.3333.02-00  0x00000007   0x8C30 1199         0/0/0
0000.0000.7500.00-00  0x0000308D   0x5EDF 1198         0/0/0
0090.1A00.B000.00-00  0x00000011   0xB082 1195         1/0/0
0090.1A00.C000.00-00  0x0000005F   0x9860 1196         0/0/0

IS-IS Level-2 Link State Database
LSPID LSP    Seq Num     LSP Checksum LSP Holdtime ATT/P/OL
0000.0000.004E.00-00  0x00001355   0x0DA7 1198         0/0/0
0000.0000.3333.00-00* 0x00000257   0x566B 1199         0/0/0
0000.0000.3333.02-00  0x00000007   0x8C30 1199         0/0/0
0000.0000.7500.00-00  0x00003315   0x3627 1198         0/0/0
0010.7B36.5FF7.00-00  0x00000BAF   0x187A 1183         0/0/0
0090.1A00.B000.00-00  0x00000016   0xD624 1195         1/0/0
0090.1A00.C000.00-00  0x00000071   0x9358 1196        0/0/0
```

- Example 2

```
host1#show isis database detail
LSPID LSP    Seq Num     LSP Checksum LSP Holdtime ATT/P/OL
boston.00-00*0x000001160x4760 6551/0/0
  Area Address: 47.0005.80FF.F800.0000.0000.0004
  NLPID:       0x81 0xcc
  IP Address:  10.1.1.1
  Hostname: boston
  Router ID:  10.1.1.1
  Metric: 10 IS newyork.00
    IPv4 Interface Address:  10.1.1.1
    IPv4 Neighbor Address:   10.1.1.2
  Metric: 10 IS washington.00
    IPv4 Interface Address:  10.1.3.1
    IPv4 Neighbor Address:   10.1.3.3
  Metric: 10 IP 192.168.1.0/24
  Metric: 10 IP 10.1.1.0/24  tag value(s):   11
  Metric: 10 IP 10.1.3.0/24
  Metric: 20 IP 10.1.2.0/24  tag value(s):   22
```

- Example 3

```
host1#show isis database verbose
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
zion.00-00*           0x00000011   0xBFAD        487           0/0/0
  Area Address: 47.0005.80FF.F800.0000.0000.0003
  NLPID:       0x81 0xcc
  IP Address:  222.9.1.1
  Hostname: zion
  Router ID:  222.9.1.1
  Metric: 0  ES 2220.0900.1001
  Metric: 10 IS london.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.1.1
        IPv4 Neighbor Address:   221.1.1.2
        Maximum link bandwidth:  50000
        Reservable link bandwidth:  50000
        Unreserved bandwidth:
          Priority 0:  50000
          Priority 1:  50000
          Priority 2:  50000
          Priority 3:  50000
          Priority 4:  30000
          Priority 5:  30000
          Priority 6:  30000
          Priority 7:  30000
        TE default metric:  0
  Metric: 10 IS london.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.6.1
        IPv4 Neighbor Address:   221.1.6.2
        Maximum link bandwidth:  50000
        Reservable link bandwidth:  50000
        Unreserved bandwidth:
          Priority 0:  50000
          Priority 1:  50000
          Priority 2:  50000
          Priority 3:  50000
          Priority 4:  30000
          Priority 5:  30000
          Priority 6:  30000
          Priority 7:  30000
        TE default metric:  0
  Metric: 10 IS paris.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.4.1
        IPv4 Neighbor Address:   221.1.4.4
        Maximum link bandwidth:  0
        Reservable link bandwidth:  0
        Unreserved bandwidth:
          Priority 0:  0
          Priority 1:  0
          Priority 2:  0
          Priority 3:  0
          Priority 4:  0
          Priority 5:  0
          Priority 6:  0
          Priority 7:  0
        TE default metric:  0
```

```
  Metric: 10 IP 221.1.1.0/24
  Metric: 10 IP 221.1.6.0/24
  Metric: 10 IP 221.1.4.0/24
  Metric: 0 IP 222.9.1.1/32
```

- Example 4

```
host1#show isis database Getafix:v2
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Getafix:v2.00-00*     0x00000001   0xEB53        1097          0/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Getafix:v2.00-00*     0x00000001   0x456D        1097          0/0/0
```

- Example 5

```
host1#show isis database Getafix:v2 detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Getafix:v2.00-00*     0x00000001   0xEB53        967           0/0/0
  Area Address: 22
  NLPID:       0x81 0xcc
  IP Address:  1.1.1.2
  Hostname: Getafix:v2
  Metric: 10 IS Getafix:v2.01
  Metric: 0  ES Getafix:v2
  Metric: 10 IP 1.1.1.0 255.255.255.0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Getafix:v2.00-00*     0x00000001   0x456D        967           0/0/0
  Area Address: 22
  NLPID:       0x81 0xcc
  IP Address:  1.1.1.2
  Hostname: Getafix:v2
  Metric: 10 IS Getafix:v2.01
  Metric: 10 IP 1.1.1.0 255.255.255.0
```

- Example 6—For IS-IS IPv6 configuration

```
host1:2#show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
host1:1.00-00          0x00000005   0x0E39        930           0/0/0
  Area Address: 49.0001
  NLPID:       0x81 0xcc
  IP Address:  4.4.4.1
  Hostname: host1:1
  Metric: 0  ES host1:1
  Metric: 10 IS host1:2.00
  Metric: 10 IS host1:2.00
  Metric: 10 IP 4.4.4.0/24
  Metric: 10 IP 20.0.0.0/24
  Metric: 10 IPv6 Internal Up 1:1:1:101::/64
host1:2.00-00*         0x00000004   0xC558        735           0/0/0
  Area Address: 49.0001
  NLPID:       0x81 0xcc
  IP Address:  9.9.9.9
  Hostname: host1:2
  Metric: 0  ES host1:2
  Metric: 10 IS host1:1.00
  Metric: 10 IS host1:3.00
```

```
Metric: 10 IS host1:1.00
Metric: 10 IS host1:3.00
Metric: 10 IP 4.4.4.0/24
Metric: 10 IP 20.0.0.0/24
Metric: 10 IP 40.0.0.0/24
Metric: 10 IP 30.0.0.0/24
Metric: 10 IPv6 Internal Up 1:1:1:102::/64
```

show isis mpls adjacency-log

- Use to display a log of the last 20 IS-IS adjacency changes.
- Field descriptions
  - When—Amount of time since recording the log entry
  - Neighbor ID—Identifier for the neighbor
  - IP Address—IP address of the neighbor
  - Interface—Interface from which neighbor was learned
  - Status—Adjacency status, Up or Down
  - Level—IS-IS routing level
- Example

```
host1#show isis mpls adjacency-log
IS-IS MPLS TE log
When      Neighbor ID        IP Address   Interface Status  Level
02:25:47  2220.0900.2002.00  221.1.1.2    at2/0.1   Up      L1
02:25:47  2220.0900.2002.00  221.1.6.2    at2/0.6   Up      L1
02:25:47  2220.0900.4004.00  221.1.4.4    at2/1.5   Up      L1
```

show isis mpls advertisements

- Use to display the last record flooded from MPLS.
- Field descriptions
  - System ID—Name or system ID of the MPLS tail-end (destination) router
  - Router ID—Router ID for the router
  - Link Count—Number of links that MPLS advertises
  - Neighbor System ID—Identifier of the remote system in an area
  - Administrative group—TLV administrative group or color assigned to the link
  - Interface IP address—IP address of the interface
  - Neighbor IP Address—IP address of the neighbor
  - Maximum link bandwidth—Bandwidth capacity of the link in bits per second
  - Reservable link bandwidth—Amount of bandwidth reservable on the link (whether reserved or not)
  - Unreserved bandwidth—Amount of bandwidth available for reservation on the link
  - TE default metric—Traffic engineering default metric value
  - Affinity Bits—Attributes flooded for the link

- Example

```
host1#show isis mpls advertisements
System ID: zion.00
  Router ID:  222.9.1.1
    Link[1]
      Neighbor System ID: london.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.1.1
        IPv4 Neighbor Address:   221.1.1.2
        Maximum link bandwidth:  50000
        Reservable link bandwidth:  50000
        Unreserved bandwidth:
          Priority 0:  50000
          Priority 1:  50000
          Priority 2:  50000
          Priority 3:  50000
          Priority 4:  30000
          Priority 5:  30000
          Priority 6:  30000
          Priority 7:  30000
        TE default metric:  0
    Link[2]
      Neighbor System ID: london.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.6.1
        IPv4 Neighbor Address:   221.1.6.2
        Maximum link bandwidth:  50000
        Reservable link bandwidth:  50000
        Unreserved bandwidth:
          Priority 0:  50000
          Priority 1:  50000
          Priority 2:  50000
          Priority 3:  50000
          Priority 4:  30000
          Priority 5:  30000
          Priority 6:  30000
          Priority 7:  30000
        TE default metric:  0
    Link[3]
      Neighbor System ID: paris.00
        Administrative group:  0
        IPv4 Interface Address:  221.1.4.1
        IPv4 Neighbor Address:   221.1.4.4
        Maximum link bandwidth:  0
        Reservable link bandwidth:  0
        Unreserved bandwidth:
          Priority 0:  0
          Priority 1:  0
          Priority 2:  0
          Priority 3:  0
          Priority 4:  0
          Priority 5:  0
          Priority 6:  0
          Priority 7:  0
        TE default metric:  0
```

show isis mpls tunnel

- Use to display information about tunnels used in the calculation of IS-IS next hops.
- Field descriptions
  - System Id—Name or system ID of the MPLS tail-end (destination) router
  - Tunnel Name—Name of the MPLS tunnel interface
  - Nexthop—Destination IP address of the MPLS tunnel
  - Metric —Metric of the MPLS tunnel
  - Mode—Metric mode, either absolute or relative
- Example

```
host1#show isis mpls tunnel
System Id         Tunnel Name   Nexthop    Metric    Mode
dakota-router1.00 Tunnel1       2.2.2.2    -3        Relative
                  Tunnel2       2.2.2.2    11        Absolute
jersey-router2.00 Tunnel3       3.3.3.3    -1        Relative
                  Tunnel4       3.3.3.3
```

show isis nsf

- Use to display information about the configured and operational settings on the router for IS-IS graceful restart, which is also known as nonstop forwarding (NSF).
- Field descriptions
  - Configured Timer Values—Displays the following values configured for IS-IS graceful restart on the router, as described in *Configuring Graceful Restart* on page 365:
    - Graceful Restart—Setting for IS-IS graceful restart on the router: Enabled or Disabled
    - T3 Timer—Method by which the restarting router obtains the T3 wait time: Manual or Derive from adjacency
    - T3 Timeout Value—Maximum time, in seconds, that the restarting router waits before setting the overload bit to indicate that IS-IS graceful restart has failed
    - T2 Timeout Value—Maximum time for IS-IS level 1 routing and level 2 routing, in seconds, that the restarting router waits for the LSP database to synchronize
    - T1 Timeout Value—Time interval, in seconds, between IS-IS restart requests sent by the restarting router on this interface to neighboring routers
    - T1 Retry Count—Number of times the restarting router resends unacknowledged restart requests on this interface at the specified interval
    - Adj. Wait Time—Maximum time, in seconds, that an IS-IS process on the restarting router waits for all interfaces with IS-IS adjacencies to come up before completing the restart process

- Operation Timer Values—Displays the following currently remaining timer settings, in seconds, for IS-IS graceful restart during the restart process:
  - T3 Timer—Remaining time before the restarting router sets the overload bit to indicate that graceful restart has failed
  - T2 Timeout Value—Remaining time for level 1 routing and level 2 routing that the restarting router waits for the LSP database to synchronize
  - Adj. Wait Time—Remaining time that the restarting router waits for all adjacencies to come up before completing the restart process
  - Restart Ack Recv Adj Count—Number of neighboring IS-IS routers for level 1 routing and level 2 routing that have acknowledged the restart requests sent by the router
  - LAN If DIS Wait Count—Number of interfaces on which the restarting router is waiting to receive election of the designated intermediate system (DIS)
  - Restart CSNP Adj Recv Count—Number of adjacencies for level 1 routing and level 2 routing that have sent complete sequence number PDUs (CSNPs) to provide information about LSP database synchronization
  - Local LSP Wait Count—Number of level 1 and level 2 LSPs for which the restarting router is awaiting complete synchronization

- Example

```
host1#show isis nsf
Configured Timer Values
-----------------------
 Graceful Restart           : Enabled
 T3 Timer                   : Manual
 T3 Timeout Value           : 80
 T2 Timeout Value           : 70(level-1)
                              70(level-2)
 T1 Timeout Value           : 60
 T1 Retry Count             : 3
 Adj. Wait Time             : 30

Operation Timer Values
----------------------
 T3 Timer                   : 0
 T2 Timeout Value           : 0(level-1)
                              0(level-2)
 Adj. Wait Time             : 0
 Restart Ack Recv Adj Count : 0(level-1)
                              0(level-2)
 LAN If DIS Wait Count      : 0
 Restart CSNP Adj Recv Count: 0(level-1)
                              0(level-2)
 Local LSP Wait Count       : 0(level-1)
                              0(level-2)
```

show isis spf-log

- Use to display how often and why the router has run a full SPF calculation.
- Field descriptions
  - When—Amount of time since a full SPF calculation took place, given in hours:minutes:seconds. The previous 20 calculations are logged.
  - Duration—Number of seconds to complete this SPF run. The elapsed time is in actual clock time, not CPU time.
  - First Trigger LSP—Whenever a full SPF calculation is triggered by a new LSP, the LSP ID is stored in the router
  - SpfType—Type of SPF run
  - Triggers—List of causes that triggered the SPF calculation
- Example 1

```
host1#show isis spf-log
Level 1 SPF log
When     Duration      First Trigger LSP    SpfType  Triggers
00:01:45 0.000         0000.0000.0000.00-00 Full     LSP Add
00:01:36 0.000         0000.0000.0000.00-00 Full     LSP Add
00:01:31 0.000         0000.0101.0101.00-00 Full     LSP Add
00:00:08 0.000         0000.0101.0101.00-00 PRC      LSP Sequence Update
```

- Example 2

```
host1#show isis spf-log detail
Level 1 SPF log
When     Duration      First Trigger LSP    SpfType  Triggers
00:01:53 0.000         0000.0000.0000.00-00 Full     LSP Add
  RTupdt 0.000
  RtLeak 0.000
00:01:44 0.000         0000.0000.0000.00-00 Full     LSP Add
  RTupdt 0.000
  RtLeak 0.000
00:01:39 0.000         0000.0101.0101.00-00 Full     LSP Add
  RTupdt 0.000
  RtLeak 0.000
00:00:16 0.000         0000.0101.0101.00-00 PRC      LSP Sequence Update
  RTupdt 0.000
  RtLeak 0.000
```

show isis summary-addresses

- Use to display the status of IS-IS aggregate addresses.
- Field descriptions
  - Address—Aggregate addresses advertised by summarization process
  - Mask—IP subnet masks used for the summary routes
  - Level—Level for which multiple groups of addresses can be summarized
  - Metric—Metric used to advertise the summary
  - State—State of the summary address
  - Prefix—IPv6 prefix
  - Tag—Number in the range 1–4294967295 that identifies the route tag assigned to the IS-IS IPv6 interface

- Example 1—For IS-IS IP addresses

```
host1#show isis summary-addresses
Address       Mask          Level       Metric   State
--------      ----------    ---------   ------   -------
3.0.0.0       255.0.0.0     LEVEL-1     0        ENABLED
4.0.0.0       255.0.0.0     LEVEL-1-2   5        ENABLED
```

- Example 2—For IS-IS IPv6 addresses

```
host1#show isis summary-addresses
Prefix          Level            Metric         Tag     State
------------    -------------    ----------     ------  -------
2008::0/8       LEVEL-2          0              100     ENABLED
```

show isis topology

- Use to display the paths to all intermediate systems or specific types of intermediate systems.
- Field descriptions
  - System ID—Name or system ID of the intermediate system
  - Metric—Metric of the path to the intermediate system
  - Next Hop—Destination IP address of the intermediate system
  - Interface—Interface from which neighbor was learned
  - SNPA—Subnetwork point of attachment; for a LAN circuit, it is the MAC address; not meaningful for a point-to-point circuit.
- Example

```
host1#show isis topology level-1
              IS-IS paths for level-1 routers
              -------------------------------
  System-ID   Metric    Next Hop       Intf      SNPA
------------- ------   -------------  --------   ----
barcelona:vr2  10      barcelona:vr2  at2/0.12
```

undebug isis

- Use to cancel the display of information about a selected event.
- The same IS-IS variables can be designated as in the **debug isis** command.
- Example

  host1#**undebug isis adj-packets**

- There is no **no** version.

Displaying CLNS

You can display the following information related to the CLNS protocol:

- CLNS information about interfaces
- Information about router adjacencies
- Information about ES and IS neighbors
- Protocol-specific information for each routing process
- Information about CLNS packets
- Global CLNS configurations

You can set a statistics baseline for CLNS using the **baseline clns** command.

baseline clns

- Use to set a statistics baseline for CLNS.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the optional *interface-specifier* parameter to specify an interface; otherwise the command sets a baseline for all interfaces.
- You cannot set a baseline for groups of interfaces.
- When baselining is requested, the time since the last baseline was set is displayed in days, hours, minutes, and seconds.
- Use the optional **delta** keyword with the **show clns traffic** command to specify that baselined statistics are to be shown.
- Example

```
host1#show clns traffic detail
IS-IS: Baseline last set 0 days, 0 hours, 1 minutes, 41 seconds
IS-IS: Corrupted LSPs: 0
IS-IS: L1 LSP Database Overloads: 0
IS-IS: L2 LSP Database Overloads: 0
IS-IS: Area Addresses Dropped: 0
IS-IS: Attempts to Exceed Max Sequence: 0
IS-IS: Sequence Numbers Skipped: 6
IS-IS: Own LSPs Purged: 1
IS-IS: System ID Length Mismatches: 0
IS-IS: Maximum Area Mismatches: 0

Interface: atm2/1.3
IS-IS: Baseline last set 0 days, 0 hours, 1 minutes, 43 seconds
IS-IS: Protocol PDUs (in/out): 32/36
IS-IS: Init Failures: 0
IS-IS: Adjacencies Changes: 1
IS-IS: Adjacencies Rejected: 0
IS-IS: Bad LSPs: 0
IS-IS: Level-1 Designated IS Changes: 0
IS-IS: Level-2 Designated IS Changes: 0
IS-IS: Invalid 9542s: 0
host1#baseline clns atm 2/1.3
```

```
host1#show clns traffic detail delta
IS-IS: Baseline last set 0 days, 0 hours, 2 minutes, 27 seconds
IS-IS: Corrupted LSPs: 0
IS-IS: L1 LSP Database Overloads: 0
IS-IS: L2 LSP Database Overloads: 0
IS-IS: Area Addresses Dropped: 0
IS-IS: Attempts to Exceed Max Sequence: 0
IS-IS: Sequence Numbers Skipped: 6
IS-IS: Own LSPs Purged: 1
IS-IS: System ID Length Mismatches: 0
IS-IS: Maximum Area Mismatches: 0

Interface: atm2/1.3
IS-IS: Baseline last set 0 days, 0 hours, 0 minutes, 8 seconds
IS-IS: Protocol PDUs (in/out): 2/1
IS-IS: Init Failures: 0
IS-IS: Adjacencies Changes: 0
IS-IS: Adjacencies Rejected: 0
IS-IS: Bad LSPs: 0
IS-IS: Level-1 Designated IS Changes: 0
IS-IS: Level-2 Designated IS Changes: 0
IS-IS: Invalid 9542s: 0
```

- There is no **no** version.

clear isis database

- Use to delete all entries from the IS-IS link-state database, or only the entries associated with the specified neighbor.
- Field descriptions
  - LSPID—LSP identifier
  - LSP Seq Num—Sequence number for the LSP. Enables other routers to determine if they have received the latest information from source.
  - LSP Checksum—Checksum of the LSP packet
  - LSP Holdtime—Number of seconds that the LSP remains valid
  - ATT—Attach bit; indicates that the router is a level 2 router and can reach other areas
  - P—P bit; detects whether intermediate system is capable of area partition repair
  - OL—Overload bit; determines whether intermediate system is congested
- Example

```
host1#show isis database
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
rtr1.00-00*    0x00000009   0x568F        1188          0/0/0
rtrtwo.00-00   0x00000005   0xEC9B        444           0/0/0

IS-IS Level-2 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
rtr1.00-00*    0x00000010   0xF630        1193          0/0/0
rtrtwo.00-00   0x0000000C   0xF8DA        1188          0/0/0
```

```
host1#clear isis database
host1#show isis database
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL

IS-IS Level-2 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
```

- There is no **no** version.

show clns

- Use to display global CLNS information about the router.
- Field descriptions
  - Interfaces Enabled for CLNS—Number of interfaces that have the CLNS routing protocol enabled
  - Configuration Timer—Interval (in seconds) after which the router sends out IS hello packets
  - Default Holding Timer—Length of time (in seconds) that hello packets are remembered
  - Packet Lifetime—Default value used in packets sourced by this router
  - Intermediate system operation enabled (forwarding allowed)—Indicates whether this router is configured to be an ES or an IS
  - IS-IS Level-1-2 Router—Shows whether IS-IS is running in this router, gives tag information, and shows whether it is running level 1 or level 1-2
  - Routing for Area—ISO (NSAP) address for the network
  - Distribute domain wide enabled—Indicates whether distribute-domain-wide is enabled
  - Area Authentication—Displays the following fields if area authentication is enabled:
    - PSNP/CSNP PDU authentication enabled—Indicates whether authentication of level 1 PSNP packets and/or level 1 CSNP packets has been enabled by means of the **area-authentication** command
    - Key-id—Numeric identifier for the authentication key
    - Type—Type of authentication: hmac-md5 or password; an asterisk after the type indicates that the key is active
    - Start Accept—Date and time that the router starts accepting packets created with this password
    - Start Generate—Date and time that the router starts inserting this password into packets
    - Stop Accept—Date and time that the router stops accepting packets created with this password
    - Stop Generate—Date and time that the router stops inserting this password into packets

- Domain Authentication—Displays the following fields if domain authentication is enabled:
  - PSNP/CSNP PDU authentication enabled—Indicates whether authentication of level 2 PSNP packets and/or level 1 CSNP packets has been enabled by means of the **domain-authentication** command
  - Key-id—Numeric identifier for the authentication key
  - Type—Type of authentication: hmac-md5 or password; an asterisk after the type indicates that the key is active
  - Start Accept—Date and time that the router starts accepting packets created with this password
  - Start Generate—Date and time that the router starts inserting this password into packets
  - Stop Accept—Date and time that the router stops accepting packets created with this password
  - Stop Generate—Date and time that the router stops inserting this password into packets

- Use the **es-neighbors** keyword to display information for IS-IS end-system adjacencies or the **is-neighbors** keyword to display information for IS-IS intermediate-system adjacencies. Neighbor entries are sorted according to the area in which they are located. The following fields are displayed when any of these keywords is used:
  - System Id—Six-byte value of router
  - Interface—Interface on which the router was discovered
  - State—Adjacency state, either Up or Init
    - Up—Believes that the ES or IS is reachable
    - Init—Router is an IS and is waiting for an IS-IS hello message. IS-IS regards the neighbor as not adjacent.
  - Type—Level 1, level 2, and level 1-2 type adjacencies
    - L1—Router adjacency for level 1 routing only
    - L1L2—Router adjacency for level 1 and level 2 routing
    - L2—Router adjacency for level 2 only
  - Priority—IS-IS priority that the respective neighbor is advertising. The highest-priority neighbor becomes the designated IS-IS router for the interface.
  - Circuit Id—Neighbor's idea of what the designated IS-IS router is for the interface
- Add the **detail** keyword to display area addresses and IP addresses.
- Example 1—For IS-IS IP configuration

```
host1#show clns
Global CLNS Information:
  3 Interfaces Enabled for CLNS
  NET: 47.0005.80FF.F800.0000.0001.0001.0000.0000.3333.00
  Configuration Timer: 10, Default Holding Timer: 30, Packet Lifetime: 1200
  Intermediate system operation enabled
```

```
  IS-IS level-1-2 Router: testnet
    Routing for Area: 47.0005.80FF.F800.0000.0001.0001
  Distribution domain wide enabled
  Area Authentication:
  PSNP PDU authentication enabled
    Key-id:   1 Type: hmac-md5
      Start Accept:   FRI JAN 14 09:57:41 2000
      Start Generate: FRI JAN 14 09:59:41 2000
      Stop Accept:    0
      Stop Generate:  0
  Domain Authentication:
  PSNP PDU authentication enabled
  CSNP PDU authentication enabled
    Key-id:   1 Type: hmac-md5*
      Start Accept:   WED JAN 12 19:01:52 2000
      Start Generate: WED JAN 12 19:03:52 2000
      Stop Accept:    0
      Stop Generate:  0
```

- Example 2—For IS-IS IPv6 configuration

```
host1:2#show clns
Global CLNS Information:
  3 Interfaces Enabled for CLNS
  NET: 49.0001.0040.0400.4002.00
  Configuration Timer: 10, Default Holding Timer: 30, Packet Lifetime: 30
  Intermediate system operation enabled
  IS-IS level-1-2 Router:
    Routing for Area: 49.0001
Ip route-type both
```

- Example 3—For IS-IS adjacencies

```
host1#show clns is-neighbors
System Id      Interface   State  Type Priority  Circuit Id
0000.0000.7500 atm2/0.111    up     L1L2 127        0000.0000.0000.00
```

- Example 4—For detailed information on IS-IS adjacencies

```
host1#show clns is-neighbors detail
System Id      Interface  State Type Priority  Circuit Id
0000.0000.7500 atm2/0.111 up    L1L2 127       0000.0000.0000.00
  Area Address(es): 47.0005.80FF.F800.0000.0001.0001
  Ip Address(es):   172.30.245.33
```

show clns interface

- Use to display CLNS-specific information about each interface.
- Field descriptions
  - interface—Status of interface
  - line protocol—Status of the line protocol, up or down
  - Checksums—Status of checksum, enabled or disabled
  - MTU—Maximum transmission size for a packet on this interface
  - Encapsulation—Encapsulation used by CLNP packets on this interface
  - Next ESH/ISH—When the next ES hello or IS hello is sent on this interface
  - Routing Protocol—One or more areas that this interface is in. In most cases, an interface is in only one area.

- Circuit type—Whether the interface has been configured for local routing (level-1), area routing (level-2), or local and area routing (level-1-2)
- Interface number—Number of the interface
- local circuit ID—Local circuit ID of the interface
- Authentication Level-1—If area authentication is enabled, lists key-id, type of authentication, start and stop accept times, and start and stop generate times for the key. An asterisk after the type indicates the key is active.
- Authentication Level-2—If domain authentication is enabled, lists key-id, type of authentication, start and stop accept times, and start and stop generate times for the key. An asterisk after the type indicates the key is active.
- Level 1 and level 2 metrics—Metric value for each level
- DIS priority—DIS priority value assigned to the IS-IS router at each level
- Priority—Priority value assigned to the IS-IS router at each level
- Circuit ID—Circuit ID of the IS-IS router at each level
- Number of active level 1 and level 2 adjacencies—Number of adjacencies active at each level
- Designated IS—Name of the designated IS-IS router at each level
- Next IS-IS LAN level Hello—Amount of time (in seconds) before the next IS-IS LAN level 1 or level 2 hello message occurs
- BFD—State of BFD for IS-IS, enabled or disabled
- Mesh Group—Status of the mesh group, Active or Inactive
- LDP-IGP Synchronization—Status of synchronization, Achieved or Pending; supported only for OSPFv2

- When you specify the **brief** keyword, the output includes the following fields.
  - interface—Name of the interface
  - state—State of the interface, up or down
  - level—Configured interface level, level-1, level-2, or level-1-2
  - DIS(L-1)—Level-1 designated intermediate system (DIS) in a multiaccess network
  - DIS(L-2)—Level-2 designated intermediate system (DIS) in a multiaccess network
  - l1/l2 Metric—Metric for the interface
- Example 1

```
host1#show clns interface
FastEthernet4/1 is up, line protocol is up
  Checksums Enabled, MTU 1500, Encapsulation SNAP
  Next ESH/ISH is 5 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x10, local circuit ID 0x1
    Level-1 Metric: 10, DIS Priority: 0, Priority: 64,
            Circuit ID: 0000.0000.0000.01
         Designated IS: Getafix:v2.01 (us)
    Number of active level-1 adjacencies: 0
```

```
        Level-2 Metric: 10, DIS Priority: 0, Priority: 64,
                Circuit ID: 0000.0000.0000.01
             Designated IS: Getafix:v1.01 (Not Us)
        Number of active level-2 adjacencies: 1
        Next IS-IS LAN Level-1 Hello in 7 seconds
        Next IS-IS LAN Level-2 Hello in 6 seconds
        BFD disabled
        Mesh Group Inactive
                     LDP is configured through LDP autoconfig
                     LDP-IGP Synchronization: Achieved
```

- Example 2

```
host1#show clns interface brief
                          Clns Intf brief Table
                          ---------------------

                                                                                 l1/l2
   interface      state     level        DIS(L-1)         DIS(L-2)       Metric
---------------   -----   ---------   --------------   --------------   ------
loopback1         up      level-1-2   Point to Point   Point to Point   10/10
ATM3/1.1          up      level-1-2   Point to Point   Point to Point   10/10
FastEthernet1/1   up      level-1-2   nemo:2.03        nemo:2.03        10/10
3 interfaces up in 3 interfaces
```

show clns neighbors

- Use to display information about ES and IS neighbors.
- Use the **detail** keyword to display area addresses, IP addresses, and the ES or IS neighbor's graceful restart capability and restarting state.
- Field descriptions
  - System Id—Six-byte value of router
  - SNPA—Subnetwork point of attachment, which is the data link address; not meaningful for a point-to-point circuit
  - Interface—Interface the router was learned from
  - State—State of the ES or IS
    - Init—Router is an IS and is waiting for an IS-IS hello message. IS-IS regards the neighbor as not adjacent.
    - Up—ES or IS is considered reachable
  - Holdtime(rem)—Remaining number of seconds before this adjacency entry times out
  - Type—One of the following adjacency types:
    - ES—End-system adjacency either discovered by means of the ES-IS protocol or statically configured
    - IS—Router adjacency either discovered by means of the ES-IS protocol or statically configured
    - L1—Router adjacency for level 1 routing only
    - L1L2—Router adjacency for level 1 and level 2 routing
    - L2—Router adjacency for level 2 only

- Proto—Protocol through which the adjacency was learned. Valid protocol sources include ES-IS, IS-IS, and Static.
- Area Address(es)—Area addresses of the ES or IS
- Ip Address(es)—IP addresses of the ES or IS
- Graceful Restart Capable—Whether graceful restart is enabled (yes) or disabled (no) on the ES or IS
- Neighbor Restarting—Whether the ES or IS is currently restarting: yes or no
- BFD session—State of any BFD session for this neighbor

- Example 1—For IS-IS IP configuration

```
host1#show clns neighbors detail
System Id       SNPA   Interface   State   Holdtime(rem)   Type   Proto
1111.1111.1111         A5/0.1      up      30(29)          L1L2   IS-IS
Area Address(es): 11.1111.1111.1111.1111.1111.1111
Ip Address(es): 172.100.11.1
Graceful Restart Capable: yes
Neighbor Restarting: yes
BFD session is not-up
```

- Example 2—For IS-IS IPv6 configuration

```
host1:2#show clns neighbors detail
System Id      SNPA                Interface   State  Holdtime(rem) Type Proto
host1:1         0090.1A41.081A      F1/1        up     30(25)        L1   IS-IS
  Area Address(es): 49.0001
  Ip Address(es):   4.4.4.1
  Graceful Restart Capable: no
  Neighbor Restarting: no
host1:3         0090.1A41.081C      F1/1        up     30(27)        L1   IS-IS
  Area Address(es): 49.0001
  Ip Address(es):   4.4.4.3
  Graceful Restart Capable: no
Neighbor Restarting: no
```

show clns protocol

- Use to display protocol-specific information about a routing process.
- Field descriptions
  - IS-IS Router—IS-IS router name
  - System ID—Six-byte value of router
  - IS-Type—Routing level (level 1, level 2, or both) that is enabled on the router
  - Manual area addresses—Configured area addresses
  - Routing for area address(es)—Identified for level 1 routing processes. For level 2 routing processes, lists the domain address.
  - Interfaces supported by IS-IS—Interfaces and type
  - Distance—Configured distance value
  - Redistributing—Protocols being redistributed into IS-IS

- Example

```
host1:2#show clns protocol
IS-IS Router:
  System Id: 0040.0400.4002.00  IS-Type: level-1-2
  Operational State: Up
  Manual area address(es):
        49.0001
  Routing for area address(es):
        49.0001
  Interfaces supported by IS-IS:
        loopback1 - IP
        FastEthernet1/1 - IP,IPv6
        ATM3/1.1 - IP, IPv6
  Distance: 115
  Redistributing:
        static
```

show clns traffic

- Use to display all CLNS packets the router sees.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - IS-IS: Baseline last set—Time since the baseline was set
  - IS-IS: Corrupted LSPs—Number of LSPs received with errors
  - IS-IS: L1 LSP Database Overloads—Number of overloads in level 1
  - IS-IS: L2 LSP Database Overloads—Number of overloads in level 2
  - IS-IS: Area Addresses Dropped—Number of area addresses that the router dropped
  - IS-IS: Attempts to Exceed Max Sequence—Number of sequence wraps over maximum
  - IS-IS: Sequence Numbers Skipped—Number of LSPs received out of order
  - IS-IS: Own LSPs Purged—Number of LSPs deleted
  - IS-IS: Other LSPs Purged—Number of received LSPs deleted
  - IS-IS: System ID Length Mismatches—Number of unmatched system ID lengths
  - IS-IS: Maximum Area Mismatches—Number of rejected hellos due to area mismatches
  - IS-IS: Area/Domain Authentication Failures—Number of authentication failures on received level 1 and level 2 LSP/SNPs
  - IS-IS: Level-1 LSPs Sent Rcvd Dropped—Number of level 1 LSPs sent, received, and dropped
  - IS-IS: Level-2 LSPs Sent Rcvd Dropped—Number of level 2 LSPs sent, received, and dropped
  - IS-IS: LSP checksum errors received—Number of LSP checksum errors received

- When you specify an interface, reports include the following additional fields:
  - Interface—IS-IS interface for which details are displayed
  - IS-IS: Protocol PDUs (in/out)—Number of packets in/out on interface
  - IS-IS: Init Failures—Number of rejected hellos on interface
  - IS-IS: Adjacencies Changes—Number of times adjacencies have transitioned from down to up
  - IS-IS: Adjacencies Rejected—Number of times hellos are rejected because of an incompatibility
  - IS-IS: Bad LSPs—Number of LSPs received with errors
  - IS-IS: Level-1 Designated IS Changes—Number of times the level 1 designated router has changed
  - IS-IS: Level-2 Designated IS Changes—Number of times the level 2 designated router has changed
  - IS-IS: Invalid 9542s—Number of rejected ES hello packets
  - IS-IS: Malformed PDUs received—Number of malformed packets received
  - IS-IS: Authentication Failures—Number of authentication failures on received level 1 and level 2 hello packets
- When you specify the **detail** keyword, the output includes the following additional fields that show packet statistics and LSP statistics. The hello, CSNP, and PSNP statistics are shown only when you issue the **detail** keyword. When the interface is Ethernet, L1 and L2 hello counts are displayed; otherwise the point-to-point hello count is displayed.
  - IS-IS: Level-1 Hellos (in/out/dropped)—Number of level 1 hellos received, sent, and dropped
  - IS-IS: Level-2 Hellos (in/out/dropped)—Number of level 2 hellos received, sent, and dropped
  - IS-IS: Level-1 CSNPs (in/out)—Number of level 1 CSNPs received and sent on the interface
  - IS-IS: Level-2 CSNPs (in/out)—Number of level 2 CSNPs received and sent on the interface
  - IS-IS: Level-1 PSNPs (in/out)—Number of level 1 PSNPs received and sent on the interface
  - IS-IS: Level-2 PSNPs (in/out)—Number of level 2 PSNPs received and sent on the interface
  - IS-IS: LSP Retransmissions—Number of LSPs retransmitted on the interface
- Example 1

```
host1#show clns traffic
IS-IS: Baseline last set 0 days, 21 hours, 12 minutes, 15 seconds
IS-IS: Corrupted LSPs: 0
IS-IS: L1 LSP Database Overloads: 0
IS-IS: L2 LSP Database Overloads: 0
IS-IS: Area Addresses Dropped: 0
IS-IS: Attempts to Exceed Max Sequence: 0
IS-IS: Sequence Numbers Skipped: 5
IS-IS: Own LSPs Purged: 0
IS-IS: Other LSPs Purged: 0
```

```
IS-IS: System ID Length Mismatches: 0
IS-IS: Maximum Area Mismatches: 0
IS-IS: Area/Domain Authentication Failures: 0
IS-IS: Level-1 LSPs Sent: 1 Rcvd: 6769 Dropped: 6769
IS-IS: Level-2 LSPs Sent: 1 Rcvd: 6769 Dropped: 6769
IS-IS: LSP checksum errors received: 0
```

- Example 2

```
host1#show clns traffic fastEthernet 4/0 detail
Interface: FastEthernet4/0
IS-IS: Baseline last set 5 days, 0 hours, 3 minutes, 31 seconds
IS-IS: Protocol PDUs (in/out): 10421/5862
IS-IS: Level-1 Hellos (in/out/dropped): 610046/610456/0
IS-IS: Level-2 Hellos (in/out/dropped): 610046/610456/0
IS-IS: Level-1 CSNPs (in/out): 0/0
IS-IS: Level-2 CSNPs (in/out): 0/0
IS-IS: Level-1 PSNPs (in/out): 0/0
IS-IS: Level-2 PSNPs (in/out): 0/0
IS-IS: Init Failures: 0
IS-IS: Adjacencies Changes: 1
IS-IS: Adjacencies Rejected: 0
IS-IS: Bad LSPs: 0
IS-IS: LSP Retransmissions: 0
IS-IS: Level-1 Designated IS Changes: 1
IS-IS: Level-2 Designated IS Changes: 1
IS-IS: Invalid 9542s: 0
IS-IS: Malformed PDU received: 0
IS-IS: Authentication Failures: 0
```

- Example 3

```
host1#show clns traffic detail
IS-IS: Baseline last set 5 days, 0 hours, 3 minutes, 31 seconds
IS-IS: Corrupted LSPs: 0
IS-IS: L1 LSP Database Overloads: 0
IS-IS: L2 LSP Database Overloads: 0
IS-IS: Area Addresses Dropped: 0
IS-IS: Attempts to Exceed Max Sequence: 0
IS-IS: Sequence Numbers Skipped: 0
IS-IS: Own LSPs Purged: 0
IS-IS: Other LSPs Purged: 0
IS-IS: System ID Length Mismatches: 0
IS-IS: Maximum Area Mismatches: 0
IS-IS: Area/Domain Authentication Failures: 0
IS-IS: Level-1 LSPs Sent: 1 Rcvd: 6769 Dropped: 6769
IS-IS: Level-2 LSPs Sent: 1 Rcvd: 6769 Dropped: 6769
IS-IS: LSP checksum errors received: 0

Interface: FastEthernet4/0
IS-IS: Baseline last set 5 days, 0 hours, 3 minutes, 31 seconds
IS-IS: Protocol PDUs (in/out): 10421/5862
IS-IS: Level-1 Hellos (in/out/dropped): 610046/610456/0
IS-IS: Level-2 Hellos (in/out/dropped): 610046/610456/0
IS-IS: Level-1 CSNPs (in/out): 0/0
IS-IS: Level-2 CSNPs (in/out): 0/0
IS-IS: Level-1 PSNPs (in/out):0/0
IS-IS: Level-2 PSNPs (in/out):0/0
IS-IS: Init Failures: 0
IS-IS: Adjacencies Changes: 1
IS-IS: Adjacencies Rejected: 0
IS-IS: Bad LSPs: 0
IS-IS: LSP Retransmissions: 0
```

```
IS-IS: Level-1 Designated IS Changes: 1
IS-IS: Level-2 Designated IS Changes: 1
IS-IS: Invalid 9542s: 0
IS-IS: Malformed PDU received: 0
IS-IS: Authentication Failures: 0

Interface: FastEthernet4/1
IS-IS: Baseline last set 5 days, 0 hours, 3 minutes, 31 seconds
IS-IS: Protocol PDUs (in/out): 10421/5862
IS-IS: Level-1 Hellos (in/out/dropped): 609946/610056/0
IS-IS: Level-2 Hellos (in/out/dropped): 609946/610056/0
IS-IS: Level-1 CSNPs (in/out): 0/0
IS-IS: Level-2 CSNPs (in/out): 0/0
IS-IS: Level-1 PSNPs (in/out):0/0
IS-IS: Level-2 PSNPs (in/out):0/0
IS-IS: Init Failures: 0
IS-IS: Adjacencies Changes: 1
IS-IS: Adjacencies Rejected: 0
IS-IS: Bad LSPs: 0
IS-IS: LSP Retransmissions: 0
IS-IS: Level-1 Designated IS Changes: 1
IS-IS: Level-2 Designated IS Changes: 1
IS-IS: Invalid 9542s: 0
IS-IS: Malformed PDU received: 0
IS-IS: Authentication Failures: 0
```

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V