JUNOSe™ Software
for E-series™ Routing Platforms

Link Layer
Configuration Guide

Release 9.1.x

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

Part Number: 162-01962-00, Revision A00

JUNOSe™ Software for E-series™ Routing Platforms Link Layer Configuration Guide, Release 9.1.x
Writing: Diane Florio, Bruce Gillham, Justine Kangas, Sarah Lesway-Ball, Helen Shaw, Brian Wesley Simmons, Fran Singer
Editing: Ben Mann, Fran Mues, Sonia Saruba
Illustration: Nathaniel Woodward
Cover Design: Edmonds Design

Revision History
18 April 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 the *JUNOSe™ Software for E-series™ Routing Platforms Link Layer Configuration Guide*:

Objectives

This guide provides the information you need to configure link-layer protocols.

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.
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

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

Document	Description
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)
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.

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

Document	Description
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 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

Chapter 1
Configuring ATM

This chapter introduces basic Asynchronous Transfer Mode (ATM) concepts, describes features of the ATM interfaces, and provides information for configuring ATM on E-series routers.

This chapter contains the following sections:

Overview

ATM is a high-speed networking technology that handles data in fixed-size units called cells. It enables high-speed communication between edge routers and core routers in an ATM network.

ATM Interfaces

An ATM port can have a major interface and one or more subinterfaces. An ATM subinterface is a mechanism that enables a single physical ATM interface to support multiple logical interfaces. Several logical interfaces can be associated with a single physical interface.

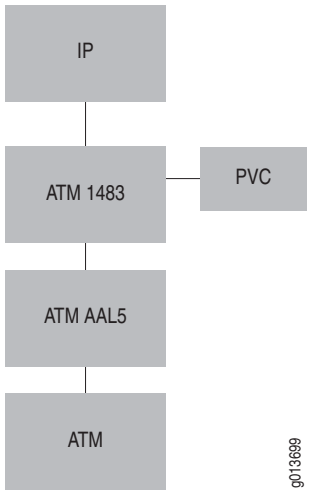
ATM subinterfaces meet the specifications in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999), which replaces RFC 1483. All references to ATM subinterfaces in this chapter are still to ATM 1483 subinterfaces.

ATM 1483 subinterfaces are identified by user-defined numbers. To select a subinterface, you append a subinterface number to the port-level **interface atm** command.

When you create an ATM 1483 subinterface, you must configure a permanent virtual circuit (PVC). Protocols such as ATM require one or more virtual circuits over which data traffic is transmitted to higher layers in the protocol stack.

Figure 1 shows a typical point-to-point ATM interface column.

Figure 1: ATM Interface Column



ATM Physical Connections

ATM interfaces and subinterfaces support two types of connections—*point-to-point* and *multipoint*. The router defaults to point-to-point.

- Point-to-point—Indicates a standard connection; for example, connecting two ATM end stations
- Multipoint—Indicates a single-source end system connected to multiple destination end systems. Multipoint indicates a nonbroadcast multiaccess (NBMA) interface. See *ATM NBMA* on page 11.

Depending on the type of connection you choose, you can specify one or more PVCs on each interface. For a standard point-to-point ATM interface, you configure only one PVC. For NBMA ATM connections, you configure multiple circuits.

ATM Virtual Connections

A *virtual connection* (VC) defines a logical networking path between two endpoints in an ATM network. ATM *cells* travel from one point to the other over a virtual connection. An ATM cell is a package of information that is always 53 bytes in length, unlike a frame or packet, which has a variable length. An ATM cell has a cell header and a *payload*. The payload contains the user data.

The cell header includes an 8-bit virtual path identifier (VPI) and a 16-bit virtual channel identifier (VCI).

An ATM network can have two types of VCs, depending on the addressing used to switch the traffic:

- Virtual channel connection (VCC)
- Virtual path connection (VPC)

Virtual Channel Connection

A VCC uses all the addressing bits of the cell header to move traffic from one link to another. The VCC is formed by joining a series of virtual channels (VCs), which are logical circuits uniquely identified for each link of the network. On a VCC, switching is done based on the combined VPI and VCI values.

Virtual Path Connection

A VPC uses the higher-order addressing bits of the cell header to move traffic from one link to another. A VPC carries many VCCs within it. A VPC can be set up permanently between two points, and then switched.

VCCs can be assigned within the VPC easily and quickly. The VPC is formed by joining a series of virtual paths, which are the logical groups of circuits uniquely defined for each link of the network. On a VPC, switching is done based on the VPI value only.

ATM SVCs

JUNOSe software does not support configuration and monitoring of ATM switched virtual circuits (SVCs) on the router.

ATM Adaptation Layer

The ATM Adaptation Layer (AAL) defines the conversion of user information into cells by segmenting upper-layer information into cells at the transmitter and reassembling them at the receiver. AAL1 and AAL2 handle intermittent traffic, such as voice and video, and are not relevant to the router. AAL3/4 and AAL5 support data communications by segmenting and reassembling packets.

E-series routers support the following AAL5 encapsulation types as specified in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999), which replaces RFC 1483:

- aal5snap—LLC/SNAP
- aal5mux ip—VC-based multiplexing
- aal5autoconfig—LLC/SNAP or VC-based multiplexing. (See *Chapter 15, Configuring Dynamic Interfaces*.)
- aal5all—Martini encapsulation

NOTE: The E120 router and the E320 router do not support Martini encapsulation (aal5all) in the current release.

Local ATM Passthrough

E-series routers support local ATM passthrough for ATM layer 2 services over Multiprotocol Label Switching (MPLS). Local ATM passthrough enables the router to emulate packet-based ATM switching. The ATM passthrough feature is useful for customers who run IP in most of their network but still have to carry a small amount of native ATM traffic.

Local ATM passthrough uses ATM Martini encapsulation to emulate ATM switch behavior. You can create pairs of cross-connected ATM VCs within the router. The router then passes AAL5 traffic between two VCs, regardless of the contents of the packets.

You can also use AAL0 encapsulation when you configure a local ATM passthrough connection. AAL0 encapsulation causes the router to receive raw ATM cells on this circuit and to forward the cells without performing AAL5 packet reassembly.

For more information, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.

VCC Cell Relay Encapsulation

E-series routers support virtual channel connection (VCC) cell relay encapsulation for ATM layer 2 services over MPLS. VCC cell relay encapsulation is useful for voice-over-ATM applications that use AAL2-encapsulated voice transmission.

VCC cell relay encapsulation enables the router to emulate ATM switch behavior by forwarding individual ATM cells over an MPLS pseudowire (also referred to as an MPLS tunnel) created between two ATM VCCs, or as part of a local ATM passthrough connection between two ATM 1483 subinterfaces on the same router. The E-series implementation conforms to the required N-to-1 cell mode encapsulation method described in the Martini draft, Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration), with the provision that only a single ATM virtual circuit (VC) can be mapped to an MPLS tunnel.

For more information, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS.*

NOTE: The E120 router and the E320 router do not support ATM over MPLS with VCC cell relay encapsulation in the current release.

Traffic Management

The scheduling priority for traffic classes depends on the type of router that you have. Table 4 describes the scheduling priorities for each type of router.

Table 4: Scheduling Priorities for Traffic Classes

Scheduling Priority (from Highest to Lowest)	ERX-7xx Models, ERX-14xx Models, or the ERX-310 Router	E120 Router and E320 Router
1	The following traffic classes are prioritized equally: ■ CBR ■ VBR-RT	CBR
2	The following traffic classes are prioritized equally: ■ VBR-NRT ■ UBR with a peak cell rate (PCR)	VBR-RT
3	UBR without PCR	VBR-NRT
4	–	UBR with or without PCR

The level of support for traffic management depends on the specific I/O module or IOA. See *Supported Features* on page 10.

Connection Admission Control

ATM networks use connection admission control (CAC) to determine whether to accept a connection request, based on whether allocating the connection's requested bandwidth causes the network to violate the traffic contracts of existing connections. CAC is a set of actions that the network takes during connection setup or renegotiation.

The router supports CAC on PVCs on major ATM interfaces. This implementation of CAC determines available bandwidth based on port subscription bandwidth. The router maintains available bandwidth for each major ATM port. Bandwidth for VP tunnels is included in CAC computations.

Table 5 lists the traffic parameter that the router uses for each service category to compute the bandwidth that the connection requires. For example, the peak cell rate is used to calculate how much bandwidth is required for CBR connections.

Table 5: Traffic Parameters Used to Compute Bandwidth

Service Category	Traffic Parameter Used to Calculate Required Bandwidth
CBR	PCR
VBR-RT	SCR
VBR-NRT	SCR
UBR	UBR bandwidth configured on the ATM major interface
UBR with PCR	UBR bandwidth configured on the ATM major interface

How CAC Works

With no connections, the available bandwidth is equal to the subscription port bandwidth. While connections are requested, the required bandwidth, which is based on the service category and traffic parameters of the connection, is compared against the available port bandwidth. If sufficient bandwidth is available, the router accepts the connection and updates the available port bandwidth accordingly.

Similarly, when a connection is deleted, the available port bandwidth is updated accordingly.

Configuring CAC

You enable and configure CAC on an ATM major interface using the **atm cac** command. When you enable CAC on an ATM interface, you can optionally specify a subscription bandwidth and a UBR weight:

- The subscription bandwidth can be greater than the effective port bandwidth to allow oversubscription. The default value of the subscription bandwidth is the effective bandwidth of the ATM port.
- The UBR weight enables you to limit the number of UBR connections by assigning a bandwidth or weight to each UBR or VBR with a PCR connection

CAC and ATM Bulk Configuration

You cannot configure CAC on an ATM interface on which you have created a bulk-configured virtual circuit (VC) range for use by a dynamic ATM 1483 subinterface. Conversely, you cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. The router rejects these configurations, which causes them to fail.

If you are upgrading to the current JUNOSe software release from a lower-numbered release, configurations that use CAC and bulk configuration on the same ATM interface continue to work. However, we recommend that you disable CAC on these ATM interfaces to ensure continued compatibility with future JUNOSe releases.

For information about how to use the **atm cac** command to configure CAC, see *Setting Optional Parameters* on page 23. For information about how to use the **atm bulk-config** command to create a bulk-configured VC range, see *Bulk Configuration of VC Ranges* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

ILMI

ATM interfaces support the ATM Forum integrated local management interface (ILMI), versions 3.0, 3.1, and 4.0. An important feature of ILMI is the ability to poll or send keepalive messages across the UNI. ATM interfaces always respond to such messages, which are sent by an ATM peer device. Optionally, you can configure ATM major interfaces to generate keepalive messages, a process that enables a continuous ATM-layer connectivity verification; if the ATM peer stops responding to keepalive messages, the router disables the ATM interface.

The ATM interface is not reenabled until the keepalive message's responses are received (or until the keepalive feature is disabled on the ATM port). To enable ILMI and control the generation of keepalive messages, use the **atm ilmi-enable** and **atm ilmi-keepalive** commands.

VPI/VCI Address Ranges

The VPI/VCI address ranges allowed on ATM interfaces are module dependent. Certain modules on ERX-14xx models, ERX-7xx models, or the ERX-310 router have a fixed allocation scheme, whereas others have a configurable allocation scheme. In the configurable allocation scheme, a bit range is shared across the VPI and VCI fields.

For example, if an ATM interface has a bit range of 18, and 4 bits are allocated to the VPI space, then 14 bits are left for the VCI space. The resulting numeric range is 0 to 2^n-1, where n is the number of bits for each space. Completing the example, if 4 bits were allocated for the VPI space and 14 for the VCI space, the configurable range would be 0 to 15 for VPI and 0 to 16,383 for the VCI space. To configure the bit range, use the **atm vc-per-vp** command.

See *Supported Features* on page 10 for details on how various line module and I/O modules support configurable VPI/VCI address ranges.

NOTE: The E120 router and the E320 router support the full VPI/VCI address range; therefore, it has a fixed allocation scheme.

VP Tunneling

Virtual path (VP) tunneling enables traffic shaping to be applied to the aggregation of all VCs within a single VP. Thus, VP tunnels can be used to ensure that the total traffic transmitted on a VP does not exceed the specified PCR. VP tunneling uses a round-robin algorithm to guarantee fairness among all of the VCs within the tunnel.

You can change the PCR associated with a tunnel even when VCs have already been configured on the tunnel. The individual VCs within a tunnel must be specified as UBR VCs. In other words, they may not have their own traffic-shaping parameters.

The level of support for VP tunneling is dependent on the specific I/O module. See *Supported Features* on page 10 for details.

Platform Considerations

You can configure ATM interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support ATM interfaces on ERX-14xx models, ERX-7xx 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 ATM.

For information about the modules that support ATM interfaces on theE120 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 MLPPP.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify an ATM interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 routers and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E120 router or E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about ATM interfaces, consult the following resources:

- ATM Forum—ATM User-Network Interface Specification, Version 3.0 (September 1993)
- ATM Forum—ATM User-Network Interface Specification, Version 3.1 (September 1994)
- ATM Forum—Integrated Local Management Interface (ILMI) Specifications, Versions 3.0, 3.1, and 4.0 (September 1996)
- ATM Forum—Traffic Management Specification, Version 4.0 (April 1996)
- ITU-T Draft Recommendation I.363 (AAL5 support) (January 1993)
- RFC 2390—Inverse Address Resolution Protocol (September 1998)
- RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999) (RFC 2684 obsoletes RFC 1483)
- ITU-T Recommendation I.610—B-ISDN Operation and Maintenance Principles and Functions (February 1999)

- Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration)
- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—See the Release Notes corresponding to your software release for information about maximum values

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Supported Features

This section describes ATM feature support on E-series modules.

For more information about the physical layer characteristics of the modules described in this section, including the numbering schemes, see the *JUNOSe Physical Layer Configuration Guide*.

Module Capabilities

The level of support for certain ATM capabilities varies depending on the module. Table 6 lists the specific differences in the capabilities of the modules.

The number of VP tunnels varies with the number of ports in the associated line module. For information about the maximum number of ATM VP tunnels supported per port for all line modules, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: Support for the OC3 (dual port) line module has been deprecated.

Table 6: ATM Capabilities on Line Modules and I/O Modules

Line Module	I/O Module or IOA	Number of VP Tunnels	VPI/VCI Address Range	Configurable Bit Range	Number of VCs on Each Port	ATM Circuit Traffic Management Types	VP Tunnel Traffic Management Types
OCx/STMx ATM	OC3-4 I/O 4xDS3 ATM I/O	1024	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
OCx/STMx ATM	OC12/STM4 I/O	256	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
OC3/STM1 GE/FE	OC3-2 GE APS I/O	1024	Configurable	20	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT

Table 6: ATM Capabilities on Line Modules and I/O Modules (continued)

Line Module	I/O Module or IOA	Number of VP Tunnels	VPI/VCI Address Range	Configurable Bit Range	Number of VCs on Each Port	ATM Circuit Traffic Management Types	VP Tunnel Traffic Management Types
ES2 4G LM	ES2-S1 OC3-8 STM1 ATM IOA	1 IOA per slot: 2048 2 IOAs per slot: 4096	Fixed VPI: 0–255 VCI: 0–65535	–	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT
ES2 4G LM	ES2-S1 OC12-2 STM4 ATM IOA	1 IOA per slot: 512 2 IOAs per slot: 1024	Fixed VPI: 0–255 VCI: 0–65535	–	8000 active 16,000 configured	CBR, UBR, UBR with PCR, VBR-NRT, VBR-RT	CBR, VBR-NRT

Virtual Channel Support

The number of virtual channels (VCs) that the router supports on each port varies depending on the E-series router and module you are using. For information about the maximum number of ATM VCs supported per chassis, per module, and per port, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

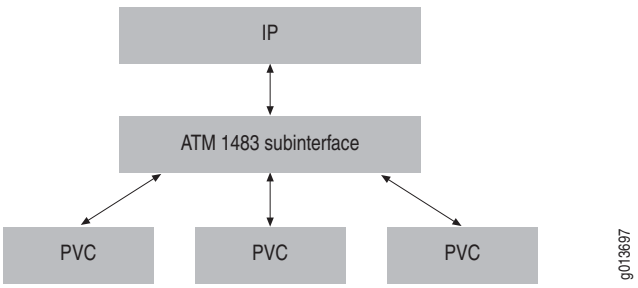
ATM NBMA

The software supports nonbroadcast multiaccess (NBMA) networks, which interconnect more than two routers and have no broadcast capabilities.

NOTE: The E120 router and the E320 router do not support ATM NBMA in the current release.

An ATM NBMA network can be thought of as an interface stack with a single IP interface at the top, eventually fanning out to multiple independent PVCs. See Figure 2.

Figure 2: NBMA Interface Stack



Unlike standard point-to-point ATM interfaces and broadcast-oriented Ethernet interfaces, NBMA interfaces form a point-to-multipoint connection. For example, you can use NBMA to connect a router to multiple stations.

An NBMA interface consists of a single ATM 1483 subinterface that has two or more VCs. You can add circuits to an existing ATM 1483 subinterface at any time. New circuits become usable after they have valid ARP table entries. NBMA circuits support only IP directly over ATM 1483.

The software restricts NBMA interfaces so that all circuits reside on the same physical interface. An NBMA interface can use as many PVCs as are available on a physical port.

ARP Table

To maintain the Address Resolution Protocol (ARP) table, you can use either static mapping via the CLI or Inverse ARP (InARP). InARP provides a way of determining the IP address of the device at the far end of a circuit. For NBMA interfaces, InARP enables automatic creation of ARP table entries for each circuit on the interface.

You must enable InARP when you create a PVC by using the **atm pvc** command. After you configure InARP, a protocol mapping between an ATM PVC and a network address is learned dynamically as a result of the exchange of InARP packets.

Static Map Versus Inverse ARP

If the device at the other end of a circuit does not support InARP, static mapping is required for that circuit. One of these two methods must be used to generate an ARP table entry for each circuit of the NBMA interface.

InARP and static mapping are complementary within an NBMA subinterface, but are not compatible with regard to individual circuits. If InARP is configured on a circuit, the corresponding virtual circuit descriptor (VCD) cannot be present in a static map applied to that interface.

Aging

ARP table entries, with the exception of those declared static, are aged out based on an aging interval defined on a subinterface basis. For the purposes of aging, entries produced via a static map are treated as static ARP table entries. InARP-generated entries are also treated as static; however, the InARP state machine automatically removes entries that cannot be successfully refreshed after three successive failed InARP requests.

Removing Circuits

If a circuit is removed, it is also removed from the ARP table, but not from the static map. If the circuit is reconfigured, a new ARP table entry is generated from the existing map entry. If the circuit uses InARP, the ARP table entry is immediately removed on removal of the circuit.

If a subinterface is removed, all associated circuits and their associated ARP table entries are removed.

Operations, Administration, and Management of ATM Interfaces

ATM interfaces support the OAM standards of the ITU, per recommendation I.610. OAM provides VC/VP integrity and fault and performance management. The E-series router supports F4 and F5 ATM OAM fault management, loopback, and continuity check (CC) cells. These cells perform fault detection and notification, loopback testing, and link integrity.

ATM uses F4 and F5 cell flows as follows:

- F4—Used in VPs
- F5—Used in VCs

ATM interfaces always generate and validate CRC-10 checksums on OAM cells.

For information about configuring OAM on the router, see the following sections:

- *Configuring OAM* on page 30
- *Configuring F5 OAM for Data PVCs* on page 48

End-to-End and Segment Endpoints

An ATM connection consists of a group of points. This OAM implementation provides management for the following points:

- Connection endpoint—The end of a VC/VP connection where the ATM cells are terminated
- Segment endpoint—The end of a connection segment

Fault Management

ATM uses two types of fault management cells to convey defect information to the endpoints of a VP/VC:

- Alarm indication signal (AIS) cells, which are used to indicate a fault to the downstream endpoint. AIS cells contain defect type and defect location fields, which optionally convey information about the type of defect detected and the location of the defect.
- Remote defect indication (RDI) cells, which are received from the remote endpoint of the VP/VC and indicate an interruption in the cell transfer capability of the VP/VC.

Connecting points in the VP/VC that detect a fault send AIS cells in the downstream direction to the endpoint of the VP/VC. Upon receipt of AIS cells, the downstream endpoint generates RDI cells in the upstream direction to alert all connecting points and the remote endpoint of an interruption in the cell transfer capability of the VP/VC.

If fault management detects a failure condition (because of arrival of AIS or RDI cells), the router disables the corresponding VC until the fault condition is no longer detected.

How the ATM Interface Handles AIS Cells

Nodes that detect a failure send AIS cells to the downstream endpoint. Because the ATM interface is an endpoint and there is no downstream neighbor to an ATM endpoint, the ATM interface never generates AIS cells. The ATM interface responds to the receipt of AIS cells as follows:

1. When an ATM interface receives a configurable number of F4 or F5 AIS cells, it enters the AIS state.

2. While in the AIS state, the ATM interface sends F4 or F5 RDI cells to the remote endpoint. It sends the RDI cells at the rate of one cell per second for as long as the AIS condition exists.

   For all RDI cells sent, the defect type and defect location fields contain the values from the received AIS cells.

3. RDI cell generation stops when one of the following conditions occurs:

   - The interface receives an F4 or F5 loopback cell or an F4 or F5 CC cell.
   - The interface does not receive an AIS cell for a configurable time period.
   - The OAM VC status field of the **show atm vc atm** command shows that the circuit is in AIS state.

How the ATM Interface Handles RDI Cells

RDI cells received from the remote endpoint of the VP/VC indicate an interruption in the cell transfer capability of the VP/VC. For example, the remote endpoint of a VC receives an F5 AIS cell, enters the AIS state, and transmits F5 RDI cells for the duration of the AIS condition. On receipt of a configurable number of F4 or F5 RDI cells, the ATM interface declares an RDI state but does not generate OAM fault management cells in response to the condition. The ATM interface leaves the RDI condition when no RDI cells have been received for a configurable time period.

The OAM VC status field of the **show atm vc atm** command shows whether the circuit is in RDI state.

Continuity Verification

CC cells provide continual monitoring of a connection on a segment or end-to-end basis. To verify the integrity of the link, you can set up a VP or VC to regularly send or receive CC cells at either the segment level or at the end-to-end level.

The CC cell source generates the CC cells, and the sink receives and processes the cells. You can set up a VP or VC as the source, the sink, or both the source and the sink. If you enable a VP or VC as a CC cell source, it generates CC cells. The VP or VC counts CC cells whether or not CC cell flow is enabled. You can enable CC cells only on data circuits, not on control circuits, such as ILMI or signaling circuits.

Activation and Deactivation Cells

To enable and disable CC cell flows, ATM OAM uses activation and deactivation cells:

- To enable a CC cell flow, the router sends activation OAM cells to the peer. The peer replies with a confirmation or denial. If the CC sink point is not activated, all received CC cells are dropped. (See *Activating CC Cell Flow* for more details.)
- To disable a CC cell flow, the router sends deactivation OAM cells to the peer. The peer replies with a confirmation or denial.

Activating CC Cell Flow

When the router sends a CC activation cell to the peer, one of the following occurs:

- If the router receives a positive response (Activation Confirmed), the VC or VP goes to CC active state, and CC is enabled on the VC or VP.
- If the router receives a negative response (Activation Req. Denied), the VC or VP goes to CC failed state, and CC is not enabled on the VC or VP.
- If the router does not receive a response within 5 seconds, it sends another activation cell. This process is repeated three times. If the router does not receive a response, it stops the activation process.

If the VC or VP is the source point, CC cell generation starts as soon as the router sends the activation request to the peer. CC cell generation stops if the CC fails, when the maximum number of retries is reached, or when the deactivation process is complete.

Deactivating CC Cell Flow

The process of sending a deactivation request is the same as for activation cells except that deactivation cells are sent instead.

Also, the **atm oam flush** command causes the router to send a deactivation request to the peer and suspend all CC operations. Therefore, we recommend that you disable CC cell generation and transmission on all VCs before issuing **atm oam flush**.

After CC Cell Flow Is Enabled

If the VC or VP is set up as the source point, the ATM interface sends one CC cell per second. CC cell generation stops if one of the following conditions occur:

- The ATM interface goes down.
- You disable OAM CC on the circuit by using the **atm pvc** command.
- The peer deactivates the OAM CC cell flow.
- You disable OAM cell reception and transmission on the ATM interface by using the **atm oam flush** command.

If the VP is set up as a CC sink point and no CC cell is received for 4 seconds, the VP goes to AIS state and sends one RDI cell per second.

To view the current state of the activation or deactivation process, including statistics, use the **show atm oam** command for VPs and the **show atm vc atm interface** command for VCs.

Loopback

You can use loopback cells to verify connectivity between VP/VC endpoints, as well as segment endpoints within the VP/VC. You can use these tests to perform fault isolation over the VP/VC.

The ATM interface supports VC integrity, which generates F5 end-to-end loopback cells. It also supports ATM ping, which generates F4 and F5 segment and end-to-end loopback cells to test the reachability of an endpoint or a segment endpoint.

VC Integrity

VC integrity is used to monitor the operational status of an individual VC. VC integrity provides continuous ATM VC-layer connectivity verification by periodically sending F5 end-to-end loopback cells on individual PVCs to verify end-to-end connectivity. You can set the frequency with which loopback cells are transmitted for an individual VC.

If VC integrity is enabled, the peer ATM host must respond to the router's loopback cells, or the circuit will be disabled. The ATM interface does not reenable the circuit until it receives loopback responses or until local VC integrity is disabled.

You can set the following VC integrity parameters for an individual VC with the **oam retry** command. For more information, see **oam retry** on page 51.

- The retry frequency with which loopback cells are transmitted when the router verifies the down status of the circuit; that is, when the peer ATM host does not respond to a loopback cell
- The retry frequency with which loopback cells are transmitted when the router verifies the up status of the circuit; that is, when the ATM host resumes responding to a loopback cell
- The number of successive loopback cell responses missed before the router determines that the circuit is down
- The number of successive loopback responses received before the router determines that the circuit is up

VC integrity is a best-effort mechanism that tries to adhere to the loopback cell transmission frequency and retry frequency values configured for each VC without consuming excessive processing time on the line module. When you configure VC integrity for a large number of circuits on the line module, delays in transmitting OAM loopback cells might occur so new subscribers can connect and to maintain existing subscriber connections.

To set up the ATM interface to transmit F5 end-to-end loopback cells over a VC, use the **oam** keyword and an optional frequency with the **atm pvc** command. To send F5 segment loopback cells, use the ATM ping mechanism, described in *ATM Ping*.

F5 loopback receive and transmit statistics are available with the **show atm vc atm** command.

F4 OAM Cells

You can generate F4 loopback cells using the **atm oam** command or the ATM ping mechanism. F4 loopback receive and transmit statistics are available with the **show atm oam** command and include statistics on incoming and outgoing F4 end-to-end and segment loopback cells.

ATM Ping

With ATM ping you can verify whether a connection endpoint or segment point can be reached on a VC or VP. ATM ping uses F4 and F5 loopback cells and is supported only for data circuits and not control circuits (ILMI, signaling circuits). To generate:

- F5 segment loopback cells or end-to-end loopback cells, issue the **ping atm** command on a VC.
- F4 segment loopback cells or end-to-end loopback cells, issue the **ping atm** command on a VP.

You can specify the number of loopback cells that are sent, the location ID, and the timer value. After the interface sends the loopback cells, the timer is started and the interface waits for a response. On receiving the loopback response (or when the timer expires) the ATM interface sends the next cell. This operation is repeated for the number of cells specified.

Because F4 and F5 are OAM cells, disabling receipt and transmission of OAM cells on the ATM interface (by using the **atm oam flush** command) stops all outstanding ping operations on the ATM interface. You need to manually restart the ping operation after you enable receipt and transmission of OAM cells for the interface.

How the ATM Interface Handles Loopback Cells Received

The ATM interface responds to received F4 and F5 loopback cells as indicated in Table 7.

Table 7: Handling of F4 and F5 Loopback Cells Received

Loopback Cell Received	ATM Interface Response
F4 and F5 end-to-end loopback cells and segment loopback cells with the loopback location field set to all 1s (ones) and the loopback indication set.	Clears the loopback indication (sets it to all zeros) and loops back the received cell.
F4 and F5 segment loopback cells with the loopback location field set to all 0s (zeros) and the loopback indication set.	Resets the loopback indication and the location ID to all 1s (ones) and loops back the received cells.

Table 7: Handling of F4 and F5 Loopback Cells Received (continued)

Loopback Cell Received	ATM Interface Response
F4 and F5 end-to-end loopback cells and segment loopback cells with the loopback location field set to the loopback location ID of the ATM interface and the loopback indication set.	Clears the loopback indication and loops back the received cell without resetting the location ID.
F5 end-to-end loopback cells with the loopback location field set to a value other than all 1s and the loopback location ID of the ATM interface.	Discards the cell.
F5 segment loopback cells with the loopback location field set to other than all 1s (ones), set to all 0s (zeros), or set to the loopback location ID of the ATM interface.	Discards the cell.

Automatic Disabling of F5 OAM Services

The router automatically disables all F5 OAM fault management and VC integrity services configured on a VC when you change the administrative status of the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface from enabled to disabled.

To set the administrative status of an interface to disabled, use the **atm shutdown** command (for an ATM interface), the **atm aal5 shutdown** command (for an ATM AAL5 interface), or the **atm atm1483 shutdown** command (for an ATM 1483 subinterface). You can also use the **shutdown** command to disable the interface.

When F5 OAM is disabled, the OAM VC status field in the **show atm vc atm** command display indicates that the VC is not managed. The VC does not receive or transmit F5 OAM cells while F5 OAM is disabled. For examples of the **show atm vc atm** command display, see **show atm vc atm** on page 89.

When the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface is reenabled, the router automatically restores F5 OAM services on the associated VCs.

Rate Limiting for F5 OAM Cells

The router implements rate limiting for ATM F5 OAM cells to protect the corresponding ATM interface from denial-of-service (DoS) attacks. The interface discards control packets when the rate of control packets received exceeds the rate limit for ATM interfaces.

An ATM interface has a rate limit control that is non-configurable and always in effect; the rate limit is the same for all ATM interfaces. In addition, each ATM VC maintains its own state and statistics counters for tracking the rate. The rate limit for ATM OAM cells is approximately 5 packets per second.

For an ATM VC, the router increments the InOamCellDiscards statistics counter in the **show atm vc atm** command display to track the number of OAM cells received on this circuit that were discarded. The InOamCellDiscards counter operates on a per-circuit basis, not on a per-interface basis.

For examples of the **show atm vc atm** command display, see **show atm vc atm** on page 89.

Before You Configure ATM

Before you configure an ATM interface, verify that you have installed the physical module (such as an OC3 module) correctly. For more information about preconfiguration procedures, see the *ERX Hardware Guide* or the *E120 and E320 Hardware Guide*.

Also have the following information available:

- Interface specifiers for the ATM interfaces that you want to create

  For more information about specifying ATM interfaces and subinterfaces on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

- Virtual path and channel numbers for each virtual circuit you want to create

- IP addresses and subnet mask assignments for IP interfaces

You can configure the following types of dynamic interfaces over ATM:

- IP over static ATM 1483 (IPoA)

- IP over PPP over static ATM 1483

- IP over PPPoE over static ATM 1483

- IP over bridged Ethernet over static ATM 1483

- IP over MLPPP over static ATM 1483

- ATM 1483 over static ATM AAL5 over ATM

For information about creating these dynamic configurations, see *Chapter 15, Configuring Dynamic Interfaces*.

Configuration Tasks

The following sections describe how to perform these ATM configuration tasks:

Creating a Basic Configuration

To configure ATM, perform the following tasks. (Figure 3 on page 21 shows the relationship of Steps 1 through 3.)

1. Configure an ATM physical interface.

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

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 0/1.20**

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 15 22 aal5snap**

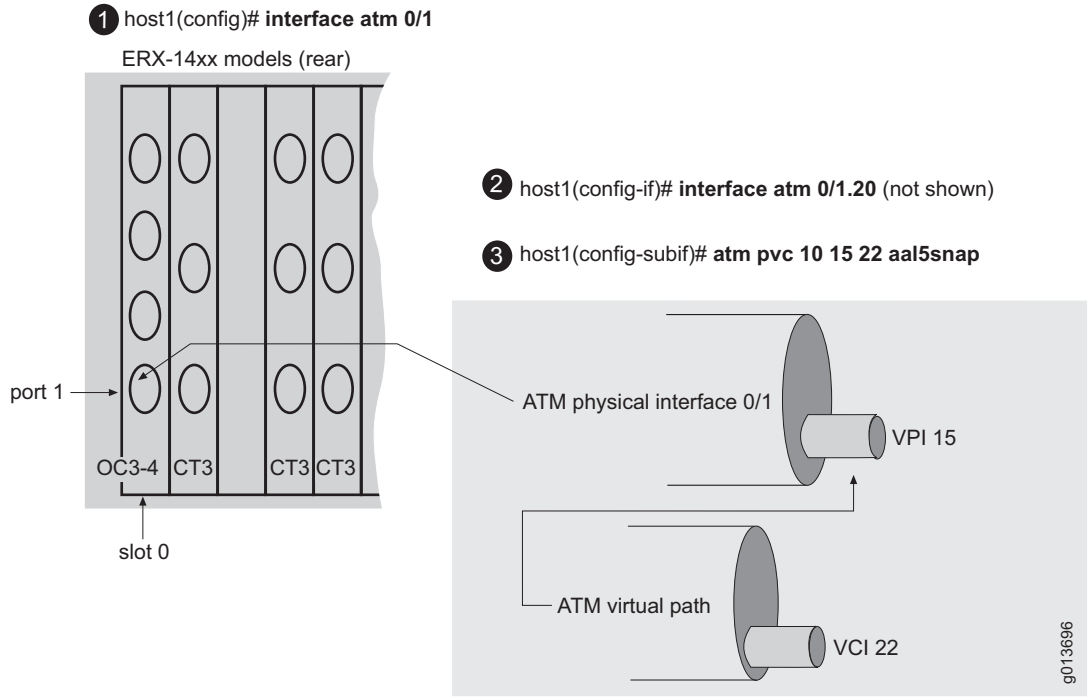
4. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#**ip address 192.32.10.20 255.255.255.0**

5. (Optional) Verify your configuration using the appropriate **show** commands.

   host1#**show atm interface atm 0/1**
   host1#**show atm vc atm 0/1 10**
   host1#**show atm subinterface atm 0/1.20**

Figure 3: Configuring an ATM Interface, Subinterface, and PVC



atm pvc

- Use to configure a PVC on an ATM interface.
- Specify one of the following encapsulation types:
  - **aal5snap**—Specifies an LLC encapsulated circuit; LLC/SNAP header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC-based multiplexed circuit. This option is used for IP only.
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed) for dynamic interfaces. See *Chapter 15, Configuring Dynamic Interfaces*, for more explanation.
  - **ilmi**—Defines the PVC for ILMI keepalive messages. You can set this option only on major interfaces. After the PVC is set up for ILMI, use the **atm ilmi-keepalive** command to cause the router to generate ILMI keepalive messages on the interface.
- You can optionally set the *peak*, *average*, and *burst* sizes. To use VBR-RT or VBR-NRT as the service type, you must specify each of these options.
- The default service type is UBR. To set a different service type, specify one of the following keywords:
  - **rt**—Selects VBR-RT as the service type. You can select **rt** only if you set the *peak*, *average*, and *burst* parameters.
  - **cbr**—Selects CBR as the service type. You must set the CBR rate in Kbps.

- To enable VC integrity and generation of OAM F5 loopback cells on this circuit, use the **oam** keyword.
- Example

  host1(config-if)#**atm pvc 6 0 11 aal5snap cbr 10000**

- Use the **no** version to remove the specified PVC.

interface atm

- Use to configure an ATM interface or subinterface type.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module; on the OC3-2 GE APS I/O module, you can specify ATM interfaces only in ports 0 and 1; port 2 is reserved for a Gigabit Ethernet interface
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for the E120 router or the E320 router, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router)
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- Specify the type of interface or subinterface: **point-to-point** or **multipoint**. Point-to-point is the default.
- Examples

  host1(config-if)#**interface atm 0/1.20**
  host1(config-if)#**interface atm 0/0/4.20**

- Use the **no** version to remove the subinterface or the logical interface.

Setting Optional Parameters

You can also set the following parameters:

- Set the administrative state of an ATM AAL5 interface to disabled.

  host1(config-if)#**atm aal5 shutdown**

- Enable CAC on the interface.

  host1(config-if)#**atm cac 3000000 ubr 3000**

- Configure the clock source.

  host1(config-if)#**atm clock internal**

- Configure framing on a T3/E3 physical interface.

  host1(config-if)#**atm framing g751adm**

- Enable ILMI on the interface.

  host1(config-if)#**atm ilmi-enable**

- Set the ILMI keepalive timer.

  host1(config-if)#**atm ilmi-keepalive 5**

- Specify the cable length (line build-out) for the ATM interface.

  host1(config-if)#**atm lbo long**

- Set the administrative state of the ATM interface to disabled.

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

- Configure SNMP link status traps on the interface.

  host1(config-if)#**atm snmp trap link-status**
  host1(config-if)#**atm aal5 snmp trap link-status**

- Set the operational mode of the physical interface to SDH STM1.

  host1(config-if)#**atm sonet stm-1**

- Configure the UNI version of ILMI using one of the following methods:

  - Enable auto configuration of ILMI.

    host1(config-if)#**atm auto-configuration**

  - Set the UNI version that the router uses when ILMI link autodetermination is unsuccessful or ILMI is disabled.

    host1(config-if)#**atm uni-version 4.0**

- Configure the number of virtual circuits for each virtual path.

  host1(config-if)#**atm vc-per-vp 128**

- Configure a virtual path tunnel and its traffic parameters.

  host1(config-if)#**atm vp-tunnel 2 128**

- Enable scrambling of the ATM cell payload on a T3 or an E3 interface.

  host1(config-if)#**ds3-scramble**

- Set the time interval at which the router records bit and packet rates.

  host1(config-if)#**load-interval 90**

- Place the interface into loopback mode for router-to-router testing.

  host1(config-if)#**loopback diagnostic**

- Disable an interface.

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

Optional Tasks on ATM 1483 Subinterfaces

You can perform the following optional tasks on ATM 1483 subinterfaces:

- Set the MTU.

  host1(config-subif)#**atm atm1483 mtu 7800**

- Configure SNMP link status traps.

  host1(config-subif)#**atm atm1483 snmp trap link-status**

- Set the administrative state of an ATM 1483 subinterface to disabled.

  host1(config-subif)#**atm atm1483 shutdown**

- Configure an advisory receive speed.

  host1(config-subif)#**atm atm1483 advisory-rx-speed 2000**

atm aal5 shutdown

- Use to set an ATM AAL5 interface administrative state to disabled.
- When you set the administrative state of the ATM AAL5 interface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.

- Example

  host1(config-if)#**atm aal5 shutdown**

- Use the **no** version to enable a disabled interface.

atm aal5 snmp trap link-status

- Use to enable SNMP link status traps on the AAL5 layer interface.
- Example

  host1(config-if)#**atm aal5 snmp trap link-status**

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

atm atm1483 advisory-rx-speed

- Use to set an advisory receive speed for an ATM 1483 subinterface. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *JUNOSe Broadband Access Configuration Guide, Chapter 12, Configuring an L2TP LAC* for additional information about the advisory receive speed.

NOTE: If you specify an advisory receive speed greater than 4294967 kbps, the speed is not accurately represented in the L2TP AVP, which is in bits per second (bps).

- The range is 0–2147483647 kbps.
- Example

  host1(config-subif)#**atm atm1483 advisory-rx-speed 2000**

- Use the **no** version to restore the default behavior—the RX speed is not sent to the LNS.

atm atm1483 mtu

- Use to set the MTU size for an ATM 1483 subinterface.
- The range is 256–9180.
- Example

  host1(config-subif)#**atm atm1483 mtu 7800**

- Use the **no** version to restore the default size of 9180.

atm atm1483 shutdown

- Use to set an ATM 1483 subinterface administrative state to disabled.
- When you set the administrative state of the ATM 1483 subinterface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.

- Example

  host1(config-subif)#**atm atm1483 shutdown**

- Use the **no** version to enable a disabled subinterface.

atm atm1483 snmp trap link-status

- Use to enable SNMP link status traps on an ATM 1483 layer subinterface.
- Example

  host1(config-subif)#**atm atm1483 snmp trap link-status**

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

atm auto-configuration

- Use to enable autoconfiguration of ILMI. Entering the **atm auto-configuration** command overrides any previous configuration of the **atm uni-version** command.
- Autoconfiguration is enabled by default.
- Example

  host1(config-if)#**atm auto-configuration**

- Use the **no** version to disable autoconfiguration and set the ILMI parameters to the UNI version configured using the **atm uni-version** command, which has a default value of UNI 4.0.

atm cac

- Use to enable CAC on the interface. You can set a subscription limit, so you can oversubscribe the port, and the UBR weight, so you can limit the number of UBR connections.
- You cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range for use by a dynamic ATM 1483 subinterface. Conversely, you cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. For information about creating bulk-configured VC ranges, see *Bulk Configuration of VC Ranges* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.
- Example

  host1(config-if)#**atm cac 3000000 ubr 3000**

- Use the **no** version to disable CAC on the interface.

atm clock internal

- Use to cause the ATM interface to generate the transmit clock internally.
- You must specify one of the following:
  - **module**—Internal clock is from the line module (the default)
  - **chassis**—Internal clock is from the configured system clock

- Example

  host1(config-if)#**atm clock internal**

- Use the **no** version to cause ATM interfaces to recover the clock from the received signal.

atm framing

- Use to configure T3 or E3 framing on an ATM interface.
- Specify one of the following framing types for a T3 (DS3) interface:
  - **cbitadm**—c-bit with ATM direct mapping
  - **cbitplcp**—c-bit with PLCP framing (default)
  - **m23adm**—M23 ATM direct mapping
  - **m23plcp**—M23 with PLCP framing
- Specify one of the following framing types for an E3 interface:
  - **g832adm**—G.832 ATM direct mapping
  - **g751adm**—G.751 ATM direct mapping
  - **g751plcp**—G.751 PLCP mapping (default)
- Example

  host1(config-if)#**atm framing g751adm**

- Use the **no** version to return framing to the default:
  - For a T3 interface, the default is **cbitplcp**
  - For an E3 interface, the default is **g751plcp**

atm ilmi-enable

- Use to enable ILMI on the interface.
- Example

  host1(config-if)#**atm ilmi-enable**

- Use the **no** version to disable ILMI on the interface.

atm ilmi-keepalive

- Use to generate ILMI keepalive messages. This value sets the time interval in seconds between poll PDU transmissions if no sequence data PDUs are pending.
- Example

  host1(config-if)#**atm ilmi-keepalive 5**

- Use the **no** version to disable the generation of keepalive messages.

atm lbo

- Use to specify the cable length (line build-out) for the ATM T3 or E3 interface. The length of cable determines power requirements.
- Specify one of the following keywords:
  - **long**—A cable length in the range 0–225 feet
  - **short**—A cable length in the range 226–450 feet (the default)
- Example

  host1(config-if)#**atm lbo long**

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

atm shutdown

- Use to set an ATM interface administrative state to disabled.
- When you set the administrative state of the ATM interface to disabled, the router automatically disables all F5 OAM services configured on the associated VC, and prevents the VC from receiving or transmitting F5 OAM cells.
- Example

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

- Use the **no** version to enable a disabled interface.

atm snmp trap link-status

- Use to enable SNMP link status traps on the ATM layer interface.
- Example

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

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

atm sonet stm-1

- Use to set the mode of operation on the physical interface to Synchronous Digital Hierarchy (SDH) Synchronous Transport Mode (STM).

  host1(config-if)#**atm sonet stm-1**

- Use the **no** version to restore the default value, SONET STS-3c operation.

atm uni-version

- Use to specify the UNI version for the interface to use.
- Valid values are 3.0, 3.1, or 4.0.
- Example

  host1(config-if)#**atm uni-version 4.0**

- There is no **no** version.

atm vc-per-vp

- Use to configure the number of VCs for each VP. The router does not execute this command when any VCs are open on the interface.
- VCs and VP tunnels must not exist when you issue this command. If they do, you must delete the VC and VP tunnel configuration before you issue this command.
- The specified value must be a power of 2, or an error message is returned.
- The minimum number of VCs per VP is 4096 for OCx/STMx ATM line modules. If you enter a value that is below the minimum, the router uses the minimum value.
- The E120 router and the E320 router support the entire VPI/VCI range; therefore, it does not support this command.
- Example

  host1(config-if)#**atm vc-per-vp 128**

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

atm vp-tunnel

- Use to define a VP tunnel and configure the rate of traffic flow within the tunnel.
- You specify a tunnel rate in Kbps. All circuits in the VP are restricted to the rate that you set.
- If any virtual circuits are open within the VPI before the tunnel is created, the router does not execute this command.
- For more information about configuring a shapeless VP tunnel for QoS, see *JUNOSe Quality of Service Configuration Guide, Chapter 19, Configuring an Integrated Scheduler to Provide QoS for ATM*.
- Example

  host1(config-if)#**atm vp-tunnel 2 128**

- Use the **no** version to remove the VP tunnel. When circuits are open within the tunnel, the router does not remove the tunnel.

ds3-scramble
e3-scramble

- Use to scramble the ATM cell payload on a T3 or an E3 interface. DS3 (T3) and E3 scrambling assists clock recovery on the receiving end of the interface.
- Example

  host1(config-if)#**ds3-scramble**

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

load-interval

- Use to set the time interval at which the router calculates bit and packet rate counters for the ATM interface.
- You can choose a multiple of 30 seconds, in the range 30–300 seconds.
- Example

  host1(config-if)#**load-interval 90**

- Use the **no** version to return to the default setting, 300 seconds.

loopback

- Use to place the interface into loopback mode.
- Specify either:
  - **diagnostic**—Places the interface into internal loopback.
  - **line**—Places the interface into external loopback.
- Example

  host1(config-if)#**loopback diagnostic**

- Use the **no** version to remove any loopback.

Configuring OAM

This section explains:

Configuring F4 OAM

The ATM interface does not support sending F4 segment loopback cells, but it does respond to F4 segment loopback cells that it receives.

F4 OAM flows need their own channel, and they are identified by the VCI on which they are sent or received. The following VCIs are reserved for F4 OAM flows for each virtual path, and you cannot open PVCs on them:

- VCI 3—For segment F4 flows
- VCI 4—For end-to-end F4 flows

NOTE: You cannot enable both loopback cells and CC cells at the same time.

To set up F4 OAM:

1. Enable F4 OAM on an interface or VP. The router enables F4 OAM at the interface level unless you specify a VPI. This example opens both segment and end-to-end F4 OAM circuits on VPI 10.

   host1(config-if)#**atm oam 10**

2. (Optional) Enable only segment or end-to-end loopback.

   host1(config-if)#**atm oam 10 seg-loopback**
   host1(config-if)#**atm oam 10 end-loopback**

3. (Optional) To cause the interface to generate end-to-end loopback cells in addition to receiving and responding to them, set the loopback timer.

   host1(config-if)#**atm oam 10 end-loopback loopback-timer 20**

4. (Optional) Enable CC cell flows.

   host1(config-if)#**atm oam 10 seg-loopback cc source**

atm oam

- Use to configure F4 OAM on an interface or circuit. F4 OAM is configured at the interface level unless you specify a VPI.
- To open F4 OAM on either a segment or end-to-end basis, use the following keywords:
  - **seg-loopback**—Enables F4 segment OAM
  - **end-loopback**—Enables F4 end-to-end OAM

NOTE: If you do not specify either segment or end-to-end loopback, the command applies to both end-to-end and segment F4 OAM circuits.

- To configure CC cell flow on the PVC, use the following keywords:
  - **both**—Enables the PVC as both the source and the sink endpoints.
  - **sink**—Enables the PVC as the sink endpoint.
  - **source**—Enables the PVC as the source endpoint.
  - **loopback-timer**—When F4 OAM is enabled, the interface or circuit accepts and responds to F4 OAM cells. However, to generate F4 loopback cells, you must configure the loopback timer in the range 1–600 seconds. This timer represents the frequency with which F4 loopback cells are transmitted. You can set the loopback timer only for end-to-end loopback.
- Example 1—Opens both F4 end-to-end and segment OAM circuits for VPI 8

  host1(config-if)#**atm oam 8**

- Example 2—Opens the F4 end-to-end OAM circuit for VPI 10 and enables sending F4 end-to-end loopback cells on the circuit at a frequency of 20 seconds

  host1(config-if)#**atm oam 10 end-loopback loopback-timer 20**

- Example 3—Opens both F4 end-to-end and segment OAM circuits on all VPs on this interface

  host1(config-if)#**atm oam**

- Example 4—Opens F4 segment OAM circuits on all VPs on this interface

  host1(config-if)#**atm oam seg-loopback**

- Example 5—Opens F4 end-to-end loopback on VPI 12

  host1(config-if)#**atm oam 12 end-loopback**

- Example 6—Opens an F4 segment OAM circuit for VPI 8 and enables CC cell generation on the segment

  host1(config-if)#**atm oam 8 seg-loopback cc source**

- Use the **no** version to delete F4 OAM circuits. Using the options, you can delete all F4 OAM circuits on the interface, segment or end-to-end F4 OAM circuits, or F4 OAM circuits on a specific VPI.

  - Example 1—Deletes all F4 OAM circuits on the interface

    host1(config-if)#**no atm oam**

  - Example 2—Deletes all F4 segment OAM circuits on the interface

    host1(config-if)#**no atm oam segment**

  - Example 3—Deletes the F4 end-to-end OAM circuit on VPI 8

    host1(config-if)#**no atm oam 8 end-loopback**

Configuring F5 OAM

F5 OAM flows run over existing PVCs. The ATM interface does not support sending F5 segment loopback cells, but it does respond to F5 segment loopback cells that it receives.

NOTE: You cannot enable both loopback cells and CC cells at the same time.

To set up F5 OAM:

1. To enable VC integrity, which causes the ATM interface to periodically send F5 end-to-end loopback cells over a VC, use the **oam** keyword with the **atm pvc** command.

   You can include the frequency (in seconds) with which the router sends F5 end-to-end loopback cells.

   host1(config-if)#**atm pvc 98 38 22 aal5snap oam 300**

2. (Optional) To enable CC cell flows on a circuit, use the **cc** keyword with the **atm pvc** command. You can enable cell flows on a segment or end-to-end basis, and you can enable the PVC as a sink, source, or both a sink and a source.

   host1(config-if)#**atm pvc 50 0 50 aal5snap oam cc end-to-end sink**

When you issue the appropriate **shutdown** command to change the administrative status of the corresponding ATM interface, ATM AAL5 interface, or ATM 1483 subinterface from enabled to disabled, the router automatically disables all F5 OAM services configured on the associated VC. For more information, see *Automatic Disabling of F5 OAM Services* on page 18.

atm pvc

- Use the **atm pvc** command with the **oam** keyword to set up the PVC to periodically transmit F5 end-to-end loopback cells over a VC.
- You can use the **oam** keyword only if you specify one of the following encapsulation types:
  - **aal5snap**
  - **aal5mux ip**
  - **aal5autoconfig**
- The **oam** keyword is not available with the **aal5all**, **aal0**, or **ilmi**
- Optionally, you can configure the time interval in the range 1–600 seconds between transmissions of OAM F5 end-to-end loopback cells.
- Use the following keywords to enable and configure CC cell flows:
  - **end-to-end**—Opens an end-to-end CC cell flow
  - **segment**—Opens a segment CC cell flow
  - **sink**—Enables this VC as a sink point (cell receiver)
  - **source**—Enables this VC as the source point (cell generator)
  - **both**—Enables this VC as both a sink point and a source point
- Example 1—Enables F5 end-to-end loopback cells

  host1(config-if)#**atm pvc 20 20 20 aal5snap oam**

- Example 2—Enables end-to-end CC cell flow and enables the PVC as the sink

  host1(config-if)#**atm pvc 5 0 5 aal5autoconfig oam cc end-to-end sink**

- Use the **no** version of the **atm pvc** command *without* the **oam** keyword to disable F5 OAM on the PVC and *without* the **cc** keyword to disable CC cell flows on the PVC. For example, the following command disables CC cell flow configured in Example 2.

  host1(config-if)#**no atm pvc 5 0 5 aal5autoconfig**

Setting a Loopback Location ID

To enable other nodes to specifically send OAM loopback cells to the ATM interface, set the location ID of the ATM interface or circuit.

host1(config-if)#**atm oam loopback-location 01090708**

NOTE: Because the router is a connection endpoint, the default loopback location ID is all 1s (ones). This command enables you to specify a nondefault value.

atm oam loopback-location

- Use to set the location ID of the ATM interface. The location ID is a 4-octet field, and the default value is all 1s (ones).
  - You can set a specific value to identify this ATM interface as the intended recipient of OAM loopback cells.
  - You can also set the location ID to all 0s (zeros).

  For information about how the router handles loopback cells based on location ID, see Table 7 on page 17.

- Example

  host1(config-if)#**atm oam loopback-location 01090708**

- Use the **no** version to return the loopback location ID to the default value, all 1s (ones).

Enabling OAM Flush

You can use the **atm oam flush** command to enable the OAM flush feature for an ATM interface. When OAM flush is enabled, the router ignores all OAM cells received on the interface, and stops sending OAM cells on this interface.

You can also issue the **atm oam flush** command with the optional **alarm-cells** keyword to cause the router to ignore only AIS and RDI cells and to accept all other OAM cells. This is useful in diagnostic situations when you might want to exclude alarm conditions.

NOTE: The OAM flush feature is supported on all E-series ATM module combinations.

atm oam flush

- Use to configure the router to ignore all OAM cells received on an ATM interface, and to stop sending OAM cells on this interface.
- To cause the router to ignore only AIS and RDI cells and to accept all other OAM cells, use the **alarm-cells** keyword.
- Example

  host1(config-if)#**atm oam flush**

- Use the **no** version to disable OAM flush on the interface.

Running ATM Ping

Keep in mind the following when you use ATM ping:

- Before you can run ATM ping, you need to add a PVC for the VPI and VCI over which you run the ping.
- Because ATM ping requires the receipt of OAM cells, make sure that the receipt and transmission of OAM cells is not disabled (using the **atm oam flush** command). To reenable the receipt and transmission of OAM cells, enter **no atm oam flush**.
- Disabling receipt of OAM cells during a ping operation stops all outstanding ping operations. You need to manually restart the ping operation after receipt of OAM cells for the interface is enabled.
- Because ATM ping is a dynamic (on-demand) operation, none of the configuration related to ATM ping is saved. To avoid acquiring excessive bandwidth for OAM, the number of outstanding ping operations on each interface is limited to 12.

ping atm interface atm

- Use to send loopback cells from an ATM interface or circuit.
- The VPI and VCI fields determine the type of loopback cells used for the ping operation. By default F5 end-to-end loopback OAM cells are used.
  - To send F4 segment loopback cells, set the VCI to 3.
  - To send F4 end-to-end loopback cells, set the VCI to 4.
- Use the **end-loopback** keyword to send the ping to the connection endpoint.
- Use the **seg-loopback** keyword to send the ping to the first segment point (for example, the next neighbor switch).

- Use the *destination* option to specify the value of the location ID included in the loopback cell. The location ID is a 16-octet field, and the destination portion is 4 octets. You can set the location ID to a specific destination or to 0s (zeros) or 1s (ones).
  - If you set the destination to 0, the loopback location ID in the loopback cell is initialized to all 0s, and each segment point in the network responds to the ping.
  - If you set the destination to 1s, the loopback location ID in the loopback cell is initialized to all 1s, and only the connection endpoint responds to the ping.
  - If you use the default value of 0xFFFFFFFF, the loopback location ID in the loopback cell is initialized to all 1s.

  For information about how the router handles loopback cells based on location ID, see Table 7 on page 17.
- The **count** keyword sets the number of OAM loopback cells to send to the destination. The default value is 5. The maximum is 32.
- The **timeout** keyword sets the amount of time to wait for a response to the sent OAM loopback cell. The default value is 5 seconds.
- The following characters can appear in the display after the **ping** command has been issued:
  - !—Each exclamation point indicates that a reply was received
  - .—Each period indicates that the ping timed out while waiting for a reply
- Example 1—This example generates end-to-end loopback cells for VPI = 0 and VCI = 105 on ATM interface 2/0. The count value is 5 OAM loopback cells, and the timeout value is 2 seconds.

  ```
  host1#ping atm interface atm 2/0 0 105 end-loopback count 5 timeout 2
  Sending 5 53-byte OAM end-to-end loopback Echoes timeout is 2 secs
  Press Ctrl+c to stop
  !!!!!
  Success rate = 100% (5/5), round-trip min/avg/max = 0/4/10 ms
  ```

- Example 2—This example generates segment loopback cells for VPI = 0 and VCI = 105 on ATM interface 2/0. The destination is set to 0xFFFFFFFF, the count value is 3 OAM loopback cells, and the timeout value is 1 second.

  ```
  host1#ping atm interface atm 2/0 0 105 seg-loopback 0xFFFFFFFF count 3
  timeout 1
  Sending 3 53-byte OAM segment loopback Echoes timeout is 1 secs
  Press Ctrl+c to stop
  !!!
  Success rate = 100% (3/3), round-trip min/avg/max = 0/3/10 ms
  ```

- There is no **no** version.

Configuring an NBMA Interface

You configure an ATM NBMA 1483 subinterface in a manner similar to configuring a standard ATM 1483 subinterface. When you specify a subinterface, however, you must select the multipoint option if you plan to add multiple circuits to form an NBMA interface. If you do not select multipoint, the subinterface defaults to point-to-point, and only a single circuit can be affiliated with that subinterface.

You can configure one or more PVCs and associate them with the subinterface you create. Also, you can enable InARP and identify a refresh rate on each specific circuit. For each NMBA interface, either InARP must be enabled, or a static map entry must be provided for each circuit owned by the interface; otherwise, transmitting over that circuit is impossible.

NOTE: NBMA interfaces support only the aal5snap encapsulation.

To configure an NBMA interface:

1. Configure a physical interface.

   host1(config)#**interface atm 2/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 2/0.2 multipoint**

3. Configure PVCs by specifying the VCD, VPI, VCI, and encapsulation type.

   host1(config-subif)#**atm pvc 1 1 1 aal5snap inarp 10**
   host1(config-subif)#**atm pvc 2 2 2 aal5snap**

4. (Optional) Specify InARP and a refresh rate (also optional).

   host1(config-subif)#**atm pvc 3 3 3 aal5snap inarp 5**
   host1(config-subif)#**atm pvc 4 4 4 aal5snap inarp**

5. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#**ip address 192.32.10.20 255.255.255.0**

6. (Optional) Use the appropriate **show** commands to verify your configuration.

   host1#**show atm interface atm 2/0**
   host1#**show atm map**
   host1#**show nbma arp atm 2/0**
   host1#**show atm vc atm 2/0 2**
   host1#**show atm subinterface atm 2/0.2**

Creating an NBMA Static Map

Static mapping creates an association between IP address–ATM PVC pairs for one or more member circuits of an ATM 1483 NBMA interface. Not every circuit necessarily gets the required association from a static map.

In the following procedure, you can repeat Step 2 for each circuit you want to map. You can associate with an interface a map group name that you have not already established. When you define the map list, the name is associated with that interface. You can perform Steps 3 and 4 before Steps 1 and 2 without affecting the results.

To set up a static map:

1. Create a map list by naming it.

   host1(config)#**map-list charlie**

2. Associate a protocol and an address with a specific virtual circuit.

   host1(config-map-list)#**ip 192.168.13.13 atm-vc 1 broadcast**

3. Specify an ATM interface.

   host1(config-if)#**interface atm 2/0**

4. Associate the map list with the interface.

   host1(config-if)#**map-group charlie**

atm pvc

- Use to configure a PVC on an ATM interface.
- InARP and refresh rate are optional parameters.
- InARP determines whether InARP requests are used and is specified on a per-circuit basis. If you disable InARP, you must use a static map table entry. Transmission over the circuit cannot occur unless you use either InARP or static map table entries.
- The default refresh rate is 15 minutes.
- You can configure InARP only if you specify the **aal5snap** encapsulation type.
- Example

  host1(config-if)#**atm pvc 6 0 11 aal5snap inarp 10**

- Use the **no** version to remove the specified PVC.

interface atm

- Use to configure an ATM interface or subinterface type.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 22.
- Specify **multipoint** to identify the subinterface as NBMA.

- Examples

  host1(config-if)#**interface atm 0/1.20**
  host1(config-if)#**interface atm 0/0/4.20**

- Use the **no** version to remove the subinterface or the logical interface.

ip atm-vc

- Use to associate a protocol and address with a specific virtual circuit.
- Use this command repeatedly for each circuit to be mapped.
- This command is available in Map List Configuration mode only.
- Example

  host1(config-map-list)#**ip 192.168.13.13 atm-vc 1 broadcast**

- Use the **no** version to remove the association.

map-group

- Use to associate the map list with an NBMA interface when configuring static mapping.
- You can issue this command before or after the **map-list** command without changing anything.
- This command is available in Interface Configuration mode only.
- See the **map-list** command.
- Example

  host1(config-if)#**map-group charlie**

- Use the **no** version to remove the association.

map-list

- Use to create a map list when configuring static mapped NBMA interfaces.
- Limit the name of the map list to no more than 31 characters.
- You can create multiple map lists; however, you can associate only one map list with each physical interface.
- If a map list contains an entry for a VCD that was previously configured to run InARP, the **map-group** command fails. If this is the case, either reconfigure the circuit with InARP disabled, or remove the entry for that circuit from the map list.
- Example

  host1(config)#**map-list charlie**

- Use the **no** version to remove the map list.

Assigning Descriptions to Interfaces

You can use the **description** commands to assign a text description or an alias to an interface, so that other **show** commands can display that information.

atm aal5 description

- Use to assign a text description or alias to an ATM AAL5 interface.
- Use the **show atm aal5 interface** command to display the text description.
- Example

  host1(config-if)#**atm aal5 description boston01**

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

atm atm1483 description

- Use to assign a text description or alias to an ATM 1483 subinterface.
- The description can be a maximum of 255 characters.
- Use the **show atm subinterface** command to display the text description.
- Example

  host1(config-subif)#**atm atm1483 description nyc33**

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

atm description

- Use to assign a text description or alias to the ATM interface.
- The description can be a maximum of 255 characters and can include the # (pound sign) character.
- The first 32 characters of the ATM description are pushed out to RADIUS during authentication and accounting.
- Use the **show atm interface** command to display the description.
- Example

  host1(config-if)#**atm description myAtm**

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

Sending Interface Descriptions to AAA

During authentication the router sends ATM interface descriptions to AAA. AAA passes the descriptions to RADIUS, and they can appear in the Calling-Station-Id attribute [31]. (For information about RADIUS and the Calling-Station-ID attribute, see *JUNOSe Broadband Access Configuration Guide, Chapter 3, Configuring RADIUS Attributes*.)

By default, the router sends the major interface descriptions to AAA on the SRP. You can configure the router to send VP interface descriptions in place of the major interface descriptions, or to send ATM 1483 subinterface descriptions to AAA on the line module. As a result, the VP or ATM 1483 subinterface descriptions can provide a convenient way to identify or group broadband access subscribers.

If you set up multiple interface descriptions, they have the following precedence:

1. ATM 1483 subinterface description
2. VP interface description
3. Major interface description

Assigning Descriptions to Virtual Paths

To assign a description to an individual VP on an ATM interface, use the **atm vp-description** command. The VP description does not affect existing descriptions configured for the ATM interface or ATM 1483 subinterface on which the VP resides. However, if you delete the ATM interface, the descriptions of all VPs residing on that interface are also deleted. In addition, if you decrease the VPI range by issuing the **atm vc-per-vp** command, the router deletes the descriptions of any VPs that are removed.

To display the VP description, use the **show atm vp-description** command, as described in *Using ATM show Commands* on page 71. Although you need not configure a VP tunnel to specify a VP description, the router also displays the VP description in the output of the **show atm vp-tunnel** command.

Exporting ATM 1483 Subinterface Descriptions

To assign a description to an ATM 1483 subinterface and configure the router to send the ATM 1483 VC interface descriptions to the line module:

1. Configure a text description for ATM 1483 subinterfaces with the **atm atm1483 description** command. This description is included in the interface identifier that is sent to AAA.

   To configure this feature for ATM 1483 subinterfaces, enter this command in Profile Configuration mode. See *Configuring ATM 1483 Dynamic Subinterfaces* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

   host1(config-subif)#**atm atm1483 description VC_atm1**

2. Set up the router to export ATM 1483 VC interface descriptions to the line module.

   host1(config)#**atm atm1483 export-subinterface-description**

3. (Optional) Display the configuration of the export ATM 1483 VC interface descriptions feature with the **show atm atm1483** command.

   ```
   host1#show atm atm1483
   ATM1483 IF Descriptions exported
   ```

4. (Optional) Display the interface descriptions with the **show atm subinterface atm** command.

atm atm1483 description

- Use to assign a text description or alias to an ATM 1483 subinterface.
- The description can be a maximum of 255 characters.
- Example

  host1(config-subif)#**atm atm1483 description nyc33**

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

atm atm1483 export-subinterface-description

- Use to export ATM 1483 VC interface descriptions to the line module. Descriptions for ATM 1483 subinterfaces are configured with the **atm atm1483 description** command.
- The description can have up to 255 characters; however, when the description is sent to the line module, it is truncated to 32 characters.
- Example

  host1(config)#**atm atm1483 export-subinterface-description**

- Use the **no** version to restore the default behavior, in which ATM 1483 interface descriptions are not exported to the line module.

atm vp-description

- Use to assign a text description to an individual VP on an ATM interface or subinterface.
- You must specify the VPI of the VP to which you want to assign the description.
- The description string can be a maximum of 32 characters.
- The VP description is stored in NVS and persists after a reboot.
- Use the **show atm vp-description** command to display the text description.
- Example

  host1(config-if)#**atm vp-description 2 vpi2Subscribers**

- Use the **no** version to restore the default value, a null string.

Configuring Individual ATM PVC Parameters

As an alternative to using the **atm pvc** command to configure ATM PVC parameters with a single command, you can access ATM VC Configuration mode to configure individual ATM PVC parameters with separate commands, one parameter at a time. You can configure parameters for the service category, encapsulation method, F5 OAM options, and Inverse ARP.

This section explains the benefits of using ATM VC Configuration mode and describes how to perform the following tasks:

Benefits

Using commands in ATM VC Configuration mode to configure individual ATM PVC parameters provides the following benefits:

- Commands in ATM VC Configuration mode are less complex and easier to use.

  With the **atm pvc** command and keywords, you configure multiple PVC attributes on a single command line. In addition, configuration attributes available only for control (ILMI and signaling) PVCs or only for data PVCs are not mutually exclusive.

  By contrast, ATM VC Configuration mode provides commands to configure each parameter individually, and makes a clearer distinction between configuration of control PVCs and configuration of data PVCs.

- ATM VC Configuration mode interoperates with the **atm pvc** command.

  You can configure all of the parameters currently supported by the **atm pvc** command from within ATM VC Configuration mode. In addition, you can create a PVC with the **atm pvc** command and modify or delete the same PVC by using ATM VC Configuration mode. Conversely, you can modify (with certain restrictions) or delete a PVC created in ATM VC Configuration mode by using the **atm pvc** command.

- ATM VC Configuration mode supports additional F5 OAM alarm surveillance and VC integrity options.

  In most cases, you can use either an ATM VC Configuration mode command or the **atm pvc** command to configure ATM PVC parameters. However, to configure F5 OAM alarm surveillance parameters (by using the **oam ais-rdi** command) or VC integrity parameters (by using the **oam retry** command), you *must* use only ATM VC Configuration mode. There are no equivalent **atm pvc** commands to configure these parameters.

  You can, however, continue to use the **atm pvc** command to enable VC integrity and modify the loopback frequency of an ATM data PVC.

NOTE: If you have existing configuration scripts that use the **atm pvc** command, we recommend that you continue to use the **atm pvc** command to configure all ATM PVC parameters except those that require you to use the **oam ais-rdi** command or **oam retry** command in ATM VC Configuration mode.

Creating Control PVCs

A control PVC, also referred to as a control circuit, supports services such as ILMI to manage and control ATM networks. You must create a control PVC on an ATM major interface, and not on an ATM 1483 subinterface that is stacked above an ATM major interface.

To create a control PVC, you issue the **pvc** command from Interface Configuration mode. However, unlike the other tasks in this section, configuring a control PVC with the **pvc** command does not access ATM VC Configuration mode.

For example, the following commands create a control PVC with VCD 10, VPI 0, VCI 16, and ILMI encapsulation.

```
host1(config)#interface atm 3/0
host1(config-if)#pvc 10 0/16 ilmi
host1(config-if)#
```

Regardless of whether you use the **pvc** command or the **atm pvc** command to create a control PVC, you cannot modify the VCD, VPI, or VCI values after they have been configured.

pvc

- Use from Interface Configuration mode to create a control PVC for Integrated Local Management Interface (ILMI).
- To create a control PVC, specify the VCD, VPI and VCI (in the format *vpi/vci*), and the **ilmi** keyword.
- Example

  host1(config-if)#**pvc 5 0/5 ilmi**

- Use the **no** version to remove the specified control PVC from the router.

Creating Data PVCs

A data PVC, also referred to as a data circuit, is an ATM PVC that carries data. You must create a data PVC on an ATM 1483 subinterface that is stacked above an ATM major interface, and not on the ATM major interface itself.

To create a data PVC, you issue the **pvc** command from Subinterface Configuration mode to access ATM VC Configuration mode. From ATM VC Configuration mode, you can then do either of the following:

- Issue the **exit** command, which creates a data PVC that uses default values for service category (unspecified bit rate without a peak cell rate), encapsulation type (**aal5snap**), F5 OAM (disabled), and Inverse ARP (disabled).
- Issue commands to configure or modify data PVC attributes including the service category, encapsulation type, F5 OAM, and Inverse ARP.

For example, the following commands create a data PVC with VCD 32, VPI 0, VCI 100 and default values for the other attributes. Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/2.2
host1(config-subif)#pvc 32 0/100
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

Regardless of whether you use the **pvc** command or the **atm pvc** command to create a data PVC, you cannot modify the VCD, VPI, or VCI values after they have been configured.

pvc

- Use from Subinterface Configuration mode to create a data PVC and access ATM VC Configuration mode, from which you can configure and modify individual PVC attributes one at a time.
- To create a basic data PVC with default values for service category, encapsulation type, F5 OAM, and Inverse ARP, specify the VCD and the VPI and VCI (in the format *vpi*/*vci*).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif)#pvc 10 15/50
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to remove the specified data PVC from the router.

Configuring the Service Category for Data PVCs

You can use individual commands in ATM VC Configuration mode to configure each supported service category on a data PVC, or to restore the default service category, unspecified bit rate (UBR) without a peak cell rate (PCR).

For example, the following commands configure a data PVC that uses the constant bit rate (CBR) service category with a nondefault PCR (10,000 Kbps). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.3
host1(config-subif)#pvc 6 0/100
host1(config-subif-atm-vc)#cbr 10000
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

cbr

- Use to configure the CBR service category on an ATM data PVC.
- You must specify a PCR, in Kbps, in the range 1–149760 (for OC3 ATM modules) or 1–599040 (for OC12 ATM modules).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#cbr 15000
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to restore the default service category, UBR without a PCR.

ubr

- Use to configure the UBR service category on an ATM data PVC.
- You can optionally specify a PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules).
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#ubr 5000
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to restore the default service category, UBR without a PCR.

vbr-nrt

- Use to configure the variable bit rate, nonreal time (VBR-NRT) service category on an ATM data PVC.
- You must specify all of the following parameters:
  - PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - SCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - Maximum burst size (MBS), in cells, in the range 0–16777215
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#**vbr-nrt 50000 10000 150**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.

vbr-rt

- Use to configure the variable bit rate, real time (VBR-RT) service category on an ATM data PVC.
- You must specify all of the following parameters:
  - PCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - SCR, in Kbps, in the range 0–149760 (for OC3 ATM modules) or 0–599040 (for OC12 ATM modules)
  - Maximum burst size (MBS), in cells, in the range 0–16777215
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#**vbr-rt 200000 30000 400**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.

Configuring Encapsulation for Data PVCs

The encapsulation method on a data PVC represents the format of the data units that traverse the circuit. You can use the **encapsulation** command in ATM VC Configuration mode to configure the encapsulation method for a data PVC, or to restore the default encapsulation method, **aal5snap**.

For example, the following commands configure a data PVC that uses **aal5all** encapsulation. Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.3
host1(config-subif)#pvc 6 0/250
host1(config-subif-atm-vc)#encapsulation aal5all
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

encapsulation

- Use to configure the encapsulation method on an ATM data PVC.
- Specify one of the following encapsulation types:
  - **aal0**—Causes the router to receive raw ATM cells on this PVC and forward the cells without performing AAL5 packet reassembly
  - **aal5all**—Configures ATM over MPLS passthrough connections; the router passes through all ATM AAL5 traffic without interpreting it
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
  - **aal5mux ip**—Configures a VC-based multiplexed circuit used for IP only
  - **aal5snap**—Configures an LLC encapsulated circuit; an LLC/SNAP header precedes the protocol datagram; this is the default encapsulation method
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#encapsulation aal5mux ip
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to restore the default encapsulation method, **aal5snap**.

Configuring F5 OAM for Data PVCs

In ATM VC Configuration mode, you can use the individual commands listed in Table 8 to configure nondefault values for F5 OAM services.

Table 8: F5 OAM Configuration Tasks and Associated Commands

To Configure	Use This Command
Surveillance parameters for alarm indication signal (AIS) and remote defect indication (RDI) fault management cells	**oam ais-rdi**
Continuity check (CC) verification	**oam cc**
Generation of F5 loopback cells and enabling of VC integrity	**oam-pvc**
Parameters for VC integrity	**oam retry**

For more information about OAM parameters, see *Operations, Administration, and Management of ATM Interfaces* on page 13.

NOTE: The **oam-ais rdi** command and the **oam retry** command are available only in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these F5 OAM alarm surveillance and VC integrity parameters.

For example, the following commands enable VC integrity on a data PVC with a nondefault loopback frequency (30 seconds). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.0
host1(config-subif)#pvc 32 0/32
host1(config-subif-atm-vc)#oam-pvc manage 30
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

The following commands, which are available only in ATM VC Configuration mode, configure nondefault VC integrity and alarm surveillance parameters on a data PVC. In this example, the VC integrity parameters configured with the **oam retry** command include the up retry count (4), down retry count (6), and retry frequency (2). The alarm surveillance parameters configured with the **oam ais-rdi** command include the alarm down count (2) and alarm clear timeout duration (4 seconds). Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/0.0
host1(config-subif)#pvc 32 0/32
host1(config-subif-atm-vc)#oam retry 4 6 2
host1(config-subif-atm-vc)#oam ais-rdi 2 4
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

oam ais-rdi

- Use to configure surveillance parameters for AIS and RDI F5 OAM fault management cells on an ATM data PVC.
- You can optionally specify the following values:
  - *alarmDownCount*—Number of successive alarm cells, in the range 1–60, for the router to receive before reporting that a PVC is down; the default value is 1
  - *alarmClearTimeout*—Number of seconds, in the range 3–60, for the router to wait before reporting that a PVC is up after the PVC has stopped receiving alarm cells; the default value is 3
- To configure these alarm surveillance parameters, you must use the **oam ais-rdi** command in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these parameters.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.

- Example

  host1(config-subif-atm-vc)#**oam ais-rdi 5 10**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default values for the alarm down count and alarm clear timeout duration.

oam cc

- Use to enable F5 OAM CC verification on an ATM data PVC.
- You can optionally specify one of the following values to configure CC cell flows:
  - **segment**—Opens an F5 OAM CC segment cell flow
  - **end-to-end**—Opens an F5 OAM CC end-to-end cell flow
- You must specify one of the following values to enable CC verification:
  - **source**—Enables this VC as the source point (cell generator)
  - **sink**—Enables this VC as a sink point (cell receiver)
  - **both**—Enables this VC as both a sink point and a source point
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example 1—Enables CC verification with a source endpoint

  host1(config-subif-atm-vc)#**oam cc source**
  host1(config-subif-atm-vc)#**exit**

- Example 2—Opens an F5 OAM CC segment cell flow and enables CC verification with a sink endpoint

  host1(config-subif-atm-vc)#**oam cc segment sink**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to disable F5 OAM CC verification and restore the default setting for cell termination, **end-to-end**.

oam-pvc

- Use to enable generation of F5 OAM loopback cells on an ATM data PVC and, optionally, enable F5 OAM VC integrity features on the circuit.
- Use this command only on data PVCs configured with **aal5snap**, **aal5autoconfig**, or **aal5 mux ip** encapsulation; the command is not valid for data PVCs configured with other encapsulation types.
- To enable F5 OAM VC integrity on the PVC, use the **manage** keyword.
- You can optionally specify the number of seconds, in the range 1–600, for the router to wait between the transmission of loopback cells during normal operation; the default value is 10.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.

- Example

  host1(config-subif-atm-vc)#**oam-pvc manage 15**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default behavior, which disables F5 OAM VC integrity on the router and restores the default value for loopback frequency, 10 seconds.

oam retry

- Use to configure F5 OAM VC integrity parameters on an ATM data PVC.
- You can optionally specify the following values:
  - *upRetryCount*—Number of successive loopback cell responses, in the range 1–60, for the router to receive before reporting that a PVC is up; default value is 3
  - *downRetryCount*—Number of successive loopback cell responses, in the range 1–60, for the router to miss before reporting that a PVC is down; default value is 5
  - *retryFrequency*—Number of seconds, in the range 1–600, for the router to wait between the transmission of loopback cells when it is verifying the state of the PVC; default value is 1
- To configure these VC integrity parameters, you must use the **oam retry** command in ATM VC Configuration mode. There is no equivalent **atm pvc** command to configure these parameters.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  host1(config-subif-atm-vc)#**oam retry 5 6 3**
  host1(config-subif-atm-vc)#**exit**

- Use the **no** version to restore the default values for the up retry count, down retry count, and retry frequency parameters.

Configuring Inverse ARP for Data PVCs

You can use the **inarp** command in ATM VC Configuration mode to enable Inverse ARP (InARP) on a data PVC that resides on an ATM 1483 NBMA subinterface configured with the **multipoint** option. The PVC must use the default encapsulation method, **aal5snap**. For more information about InARP, see *Configuring an NBMA Interface* on page 37.

For example, the following commands enable InARP with a nondefault refresh rate (10 minutes) on a data PVC. The PVC uses **aal5snap** encapsulation by default. Issuing the **exit** command causes the configuration to take effect.

```
host1(config)#interface atm 3/2.1 multipoint
host1(config-subif)#pvc 6 0/11
host1(config-subif-atm-vc)#inarp 10
host1(config-subif-atm-vc)#exit
host1(config-subif)#
```

inarp

- Use to enable Inverse ARP on an ATM PVC that resides on an ATM 1483 NBMA subinterface and uses the default encapsulation method, **aal5snap**.
- You can optionally specify an Inverse ARP refresh rate, in the range 1–60 minutes; the default value is 15.
- You must issue the **exit** command from ATM VC Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#inarp 5
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to restore the default behavior, which disables Inverse ARP on the router.

Configuring ATM VC Classes

As an alternative to configuring individual parameters for ATM data PVCs, you can access ATM VC Class Configuration mode to configure a class of attributes for an ATM data PVC. A *VC class* is a set of attributes for a virtual circuit (VC) that can include the service category, encapsulation method, F5 OAM options, and Inverse ARP.

After you configure the VC class, you then apply the attributes in the class as a group by assigning the VC class to one of the following:

- An individual PVC
- All PVCs created on a specified static ATM major interface
- All PVCs created on a specified static ATM 1483 subinterface
- A base profile from which bulk-configured VC ranges are created on a dynamic ATM 1483 subinterface

VC class assignments are valid only for ATM data PVCs created with the **pvc** command. Assigning a VC class to a PVC created with the **atm pvc** command, or to a control (ILMI) PVC, has no effect. For information about creating a data PVC by using the **pvc** command, see *Creating Data PVCs* on page 45.

NOTE: For information about the total number of VC classes supported on the router, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

Benefits

Using VC classes to configure and assign attributes to ATM data PVCs provides the following benefits:

- VC classes enable you to classify and group ATM PVCs based on the OAM and traffic requirements of their associated subscribers.

  In a typical scenario, you might group subscribers based on their OAM and traffic requirements, and then create a VC class for each subscriber group. For example, you might create two VC classes: premium-subscriber-class and economy-subscriber-class.

  In premium-subscriber-class, you might enable F5 OAM VC integrity (with the **oam-pvc manage** command), and configure a traffic class that has a higher scheduling priority, such as CBR (with the **cbr** command). Conversely, in economy-subscriber-class, you might retain the default setting that disables F5 OAM VC integrity, and configure a traffic class that has a lower scheduling priority, such as UBR with or without a PCR (with the **ubr** command). By assigning each VC class to the appropriate interfaces or individual circuits, you can group and manage the PVCs associated with the VC class based on the network requirements of the subscribers they serve.

- VC classes facilitate modifications to PVC attributes.

  If the OAM or traffic requirements change for a particular subscriber group, you can simply reconfigure the VC class associated with the PVCs for that subscriber group. This method is easier and less time-consuming than having to modify the attributes for a large number of PVCs by using individual CLI commands.

  Modifications to the attributes in a VC class affect PVCs that are already associated with this VC class as well as PVCs subsequently created for this class.

Precedence Levels

Precedence levels play an important role in determining how the router assigns the attribute values for statically created and dynamically created PVCs that have associated VC classes.

Precedence Levels for Static PVCs

For PVCs that are statically created, the router determines the PVC attribute values according to the following precedence levels, in order from highest precedence to lowest precedence:

1. The most recent explicitly set value for a PVC attribute always has the highest precedence and overrides any settings in the VC class. Explicitly set values for PVC attributes are those values configured with the CLI (by using the **atm pvc** command or commands in ATM VC Configuration mode), SNMP, or assigned by RADIUS.

2. If an attribute value is not explicitly specified, the router takes the value for that attribute from the assigned VC class, in the following order of precedence:

   a. Attribute value specified in the VC class assigned to this PVC

   b. Attribute value specified in the VC class assigned to the ATM 1483 subinterface on which this PVC is created

   c. Attribute value specified in the VC class assigned to the ATM major interface on which this PVC is created

3. If no PVC attributes are explicitly specified and no VC class assignments exist, the router applies the default values for the commands listed in Table 9 on page 56. For information about the default value for each command, see the command descriptions in *Configuring VC Classes* on page 56.

Precedence Levels for Dynamic PVCs

For PVCs that are dynamically created, the router determines the PVC attribute values according to the following precedence levels, in order from highest precedence to lowest precedence:

1. The attribute value specified in the VC class assigned in the base profile always has the highest precedence.

2. If no VC class is assigned in the base profile, the router takes the value for that attribute from the VC class assigned to the associated ATM major interface.

3. If neither the base profile nor the ATM major interface has a VC class assigned, the router takes the value for that attribute from the individually specified attributes in the base profile.

4. If neither the base profile nor the ATM major interface has a VC class assigned, and no attributes are individually specified in the base profile, the router applies the default values for the commands listed in Table 9 on page 56. For information about the default value for each command, see the command descriptions in *Configuring VC Classes* on page 56.

Precedence Level Examples

For examples that illustrate how precedence levels affect the assignment of VC classes, see *Precedence Level Examples for Assigning VC Classes* on page 64.

To help you better understand these examples, we recommend that you first read the following sections to learn how to configure and assign VC classes:

- Configuring VC Classes on page 56
- Assigning VC Classes to Individual PVCs on page 61
- Assigning VC Classes to ATM Major Interfaces on page 62
- Assigning VC Classes to Static ATM 1483 Subinterfaces on page 63
- Assigning VC Classes to Base Profiles for Bulk-Configured VC Ranges on page 63

Upgrade Considerations

The following considerations apply to using ATM VC classes when you upgrade to the current JUNOSe software release from a lower-numbered JUNOSe software release:

- It is possible to use VC classes for PVCs created in a lower-numbered release with the **atm pvc** command. In such cases, the router uses the following rules to determine the PVC attribute values:
  - Nondefault values explicitly specified for PVC attributes with the **atm pvc** command take precedence over the attribute values specified in the associated VC class. As a result, the router takes the values for these attributes from the **atm pvc** command settings.
  - Default values implicitly specified for PVC attributes with the **atm pvc** command have a lower precedence than the attribute values specified in the associated VC class. As a result, the router takes the values for these attributes from the assigned VC class.
- The output of the **show configuration** command uses either the **pvc** command format or the **atm pvc** command format to display ATM PVCs. The display format of configuration information for ATM PVCs created with the **atm pvc** command depends on the JUNOSe software release from which you are upgrading, as follows:
  - When you upgrade to the current JUNOSe software release from a JUNOSe release numbered lower than Release 7.3.x, the output of the **show configuration** command uses the **pvc** command format (**pvc** *vcd vpi*/*vci*) to display configuration information for all ATM PVCs. This occurs even if those PVCs were created in a JUNOSe release numbered lower than Release 7.3.x with the **atm pvc** command. For example, assume that you created a PVC in JUNOSe Release 7.2.x by issuing the command **atm pvc 2 0 33 aal5snap 0 0 0**. The **show configuration** command in the current JUNOSe software release displays the identifier for this PVC as follows:

    ```
    pvc 2 0/33
    ```

  - When you upgrade to the current JUNOSe software release from JUNOSe Release 7.3.x or a higher-numbered release, the output of the **show configuration** command uses the **atm pvc** command format to display configuration information for ATM PVCs created with the **atm pvc** command. For example, assume that you created a PVC in JUNOSe Release 7.3.x or Release 8.0.x by issuing the command **atm pvc 2 0 33 aal5snap 0 0 0**. The **show configuration** command in the current JUNOSe software release displays the identifier for this PVC as follows:

    ```
    atm pvc 2 0 33 aal5snap 0 0 0
    ```

    For PVCs previously created in the lower-numbered release by using the **pvc** command, the **show configuration** command displays configuration information using the **pvc** command format, as described previously.

For information about how to use the **show configuration** command, see *JUNOSe System Basics Configuration Guide, Chapter 5, Managing the System*.

To make the most efficient use of the VC class feature when you upgrade to the current JUNOSe software release, we recommend that you follow these steps:

1. Delete any PVCs created with the **atm pvc** command and recreate them by using the **pvc** command. For information about creating a data PVC by using the **pvc** command, see *Creating Data PVCs* on page 45.

2. Configure the VC class as described in *Configuring VC Classes* on page 56.

3. Assign the VC class in one of the following ways:

   - Assign the VC class to the individual PVC when you create or modify the PVC.

   - Assign the VC class to the associated ATM major interface or ATM 1483 subinterface before you create the PVC.

Configuring VC Classes

To configure a VC class, you issue the **vc-class atm** command to create and name the VC class. The **vc-class atm** command accesses ATM VC Class Configuration mode, from which you configure a set of attributes to apply to an ATM data PVC.

Table 9 lists the commands that you can use in ATM VC Class Configuration mode to configure a set of attributes for a data PVC. These commands are identical to the commands in ATM VC Configuration mode described in *Configuring Individual ATM PVC Parameters* on page 43. For more information about the syntax of each command, see the *JUNOSe Command Reference Guide A to M* and the *JUNOSe Command Reference Guide N to Z*.

Table 9: Commands to Configure VC Class Attributes

cbr	**oam-pvc**
encapsulation	**oam retry**
inarp	**ubr**
oam ais-rdi	**vbr-nrt**
oam cc	**vbr-rt**

For example, the following commands configure two VC classes: premium-subscriber-class and dsl-subscriber-class. You must issue the **exit** command from ATM VC Class Configuration mode for each VC class configuration to take effect.

```
! Configure VC class premium-subscriber-class.
host1(config)#vc-class atm premium-subscriber-class
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#cbr 200
host1(config-vc-class)#oam-pvc manage 60
host1(config-vc-class)#oam ais-rdi 5
host1(config-vc-class)#exit
```

```
! Configure VC class dsl-subscriber-class.
host1(config)#vc-class atm dsl-subscriber-class
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#ubr
host1(config-vc-class)#exit
host1(config)#
```

In premium-subscriber-class:

- The **encapsulation** command sets the encapsulation method to **aal5autoconfig**.
- The **cbr** command sets the service category to CBR with a PCR of 200 Kbps.
- The **oam-pvc** command enables generation of F5 OAM loopback cells and F5 OAM VC integrity.
- The **oam ais-rdi** command configures the alarm down count for successive AIS and RDI alarm cells to 5.

In dsl-subscriber-class:

- The **encapsulation** command sets the encapsulation method to **aal5autoconfig**.
- The **ubr** command configures the UBR service category without a PCR.

To configure an ATM VC class with systemwide default values, you can issue the **vc-class atm** command followed immediately by the **exit** command. For example, the following commands create a VC class named default-vc-class. Because no attribute values are explicitly specified in default-vc-class, the router applies the default values for the commands listed in Table 9 on page 56. For information about the default value for each command, see the command descriptions in this section.

```
! Configure VC class with default values.
host1(config)#vc-class atm default-subscriber-class
host1(config-vc-class)#exit
host1(config)#
```

To verify the VC class configuration, use the **show atm vc-class** command. For information about how to use this command, see **show atm vc-class** on page 94.

cbr

- Use to configure the CBR service category on an ATM data PVC.
- For detailed information about how to use this command, see **cbr** on page 46.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  ```
  host1(config-vc-class)#cbr 15000
  host1(config-vc-class)#exit
  ```

- Use the **no** version to restore the default service category, UBR without a PCR.

encapsulation

- Use to configure the encapsulation method on an ATM data PVC.
- For detailed information about how to use this command, see **encapsulation** on page 48.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**encapsulation aal5mux ip**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default encapsulation method, **aal5snap**.

inarp

- Use to enable Inverse ARP on an ATM PVC that resides on an ATM 1483 NBMA subinterface and uses the default encapsulation method, **aal5snap**.
- For detailed information about how to use this command, see **inarp** on page 52.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**inarp 5**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default behavior, which disables Inverse ARP on the router.

oam ais-rdi

- Use to configure surveillance parameters for AIS and RDI F5 OAM fault management cells on an ATM data PVC.
- For detailed information about how to use this command, see **oam ais-rdi** on page 49.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam ais-rdi 5 10**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default values for the alarm down count (1 successive alarm cell) and alarm clear timeout duration (3 seconds).

oam cc

- Use to enable F5 OAM CC verification on an ATM data PVC.
- For detailed information about how to use this command, see **oam cc** on page 50.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.

- Example 1—Enables CC verification with a source endpoint

  host1(config-vc-class)#**oam cc source**
  host1(config-vc-class)#**exit**

- Example 2—Opens an F5 OAM CC segment cell flow and enables CC verification with a sink endpoint

  host1(config-vc-class)#**oam cc segment sink**
  host1(config-vc-class)#**exit**

- Use the **no** version to disable F5 OAM CC verification and restore the default setting for cell termination, **end-to-end**.

oam-pvc

- Use to enable generation of F5 OAM loopback cells on an ATM data PVC and, optionally, enable F5 OAM VC integrity features on the circuit.
- For detailed information about how to use this command, see **oam-pvc** on page 50.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam-pvc manage 15**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default behavior, which disables F5 OAM VC integrity on the router and restores the default value for loopback frequency, 10 seconds.

oam retry

- Use to configure F5 OAM VC integrity parameters on an ATM data PVC.
- For detailed information about how to use this command, see **oam retry** on page 51.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**oam retry 5 6 3**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default values for the up retry count (3 successive loopback cell responses), down retry count (5 successive loopback cell responses), and retry frequency (1 second).

ubr

- Use to configure the UBR service category on an ATM data PVC.
- For detailed information about how to use this command, see **ubr** on page 46.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.

- Example

  host1(config-vc-class)#**ubr 5000**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.

vbr-nrt

- Use to configure the VBR-NRT service category on an ATM data PVC.
- For detailed information about how to use this command, see **vbr-nrt** on page 47.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**vbr-nrt 50000 10000 150**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.

vbr-rt

- Use to configure the VBR-RT service category on an ATM data PVC.
- For detailed information about how to use this command, see **vbr-rt** on page 47.
- You must issue the **exit** command from ATM VC Class Configuration mode for the configuration to take effect.
- Example

  host1(config-vc-class)#**vbr-rt 200000 30000 400**
  host1(config-vc-class)#**exit**

- Use the **no** version to restore the default service category, UBR without a PCR.

vc-class atm

- Use to create and name a VC class for an ATM data PVC.
- You must specific a VC class name of up to 32 alphanumeric characters.
- The **vc-class atm** command accesses ATM VC Class Configuration mode, from which you can configure a set of attributes for the PVC including the service category, encapsulation method, F5 OAM options, and Inverse ARP.
- You must issue the **exit** command from ATM VC Class Configuration mode for the VC class configuration to take effect.
- For information about the total number of VC classes supported on the router, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.
- Example

  host1(config)#**vc-class atm dsl-subscriber-class**
  host1(config-vc-class)#**exit**

- Use the **no** version to remove the named VC class from the router. You cannot remove a VC class that is currently assigned to at least one ATM PVC, ATM 1483 subinterface, or ATM major interface without first issuing the **no class-vc** command or the **no class-int** command to remove the VC class association with the PVC, interface, or subinterface.

Assigning VC Classes to Individual PVCs

To assign a previously configured VC class to an individual ATM data PVC, you use the **class-vc** command from ATM VC Configuration mode. Issuing this command applies the set of attributes configured in the specified VC class to the ATM data PVC.

NOTE: The **class-vc** command is valid only for a data PVC created with the **pvc** command. It has no effect for data PVCs created with the **atm pvc** command, or for control (ILMI) PVCs. For information about creating a data PVC by using the **pvc** command, see *Creating Data PVCs* on page 45.

For example, the following commands assign the VC class named premium-subscriber-class to the ATM data PVC with VCD 2, VPI 0, and VCI 200.

```
! Assign VC class premium-subscriber-class to PVC 2/0.200
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-atm-vc)#class-vc premium-subscriber-class
host1(config-subif-atm-vc)#exit
```

For those attributes that you do not explicitly specify for the ATM PVC, the router applies the values specified in the VC class. As explained in *Precedence Levels* on page 53, the values in a VC class assigned to an individual PVC take precedence over both of the following:

- Values in a VC class assigned to an ATM 1483 subinterface
- Values in a VC class assigned to an ATM major interface

For examples that illustrate how precedence levels affect the assignment of VC classes, see *Precedence Level Examples for Assigning VC Classes* on page 64.

class-vc

- Use to assign a previously configured VC class to an individual ATM data PVC.
- The **class-vc** command is valid only for data PVCs created with the **pvc** command.
- You must issue the **exit** command from ATM VC Configuration mode for the VC class association to take effect.
- Example

  ```
  host1(config-subif-atm-vc)#class-vc dsl-subscriber-class
  host1(config-subif-atm-vc)#exit
  ```

- Use the **no** version to remove the VC class association with the data PVC.

Assigning VC Classes to ATM Major Interfaces

To assign a previously configured VC class to an ATM major interface, you use the **class-int** command from Interface Configuration mode. Issuing this command applies the set of attributes in the specified VC class to the ATM data PVCs statically or dynamically created on this interface.

For example, the following commands assign the VC class named dsl-subscriber-class to an ATM major interface configured on slot 5, port 0.

```
! Assign VC class dsl-subscriber-class to ATM interface 5/0.
host1(config)#interface atm 5/0
host1(config-if)#class-int dsl-subscriber-class
host1(config-if)#exit
```

For those attributes that you do not explicitly specify for an ATM PVC, the router applies the values specified in the VC class. As explained in *Precedence Levels* on page 53, the values in a VC class assigned to an ATM major interface have a lower precedence than both of the following:

- Values in a VC class assigned to an individual ATM PVC
- Values in a VC class assigned to an ATM 1483 subinterface

This means that if a VC class is assigned to an individual PVC or ATM 1483 subinterface configured on the major interface, the attribute values configured in the VC class assigned to the PVC or subinterface override the attribute values configured in the VC class assigned to the major interface.

For examples that illustrate how precedence levels affect the assignment of VC classes, see *Precedence Level Examples for Assigning VC Classes* on page 64.

class-int

- Use from Interface Configuration mode to assign a previously configured VC class to an ATM major interface.
- You must issue the **exit** command from Interface Configuration mode for the VC class association to take effect.
- Example

  ```
  host1(config-if)#class-int gold-subscriber-class
  host1(config-if)#exit
  ```

- Use the **no** version to remove the VC class association with the interface. Issuing the **no** version causes the router to set the PVC attributes to their systemwide default values, or to the values set in the associated VC class with the next highest order of precedence.

Assigning VC Classes to Static ATM 1483 Subinterfaces

To assign a previously configured VC class to a static ATM 1483 subinterface, you use the **class-int** command from Subinterface Configuration mode. Issuing this command applies the set of attributes in the specified VC class to the ATM data PVCs statically or dynamically created on this subinterface.

For example, the following commands assign the VC class named premium-subscriber-class to an ATM 1483 subinterface configured on slot 5, port 0, subinterface 100.

```
! Assign VC class dsl-subscriber-class to ATM 1483 subinterface 5/0.100.
host1(config)#interface atm 5/0.100
host1(config-subif)#class-int premium-subscriber-class
host1(config-subif)#exit
```

For those attributes that you do not explicitly specify for an ATM PVC, the router applies the values specified in the VC class. As explained in *Precedence Levels* on page 53, the values in a VC class assigned to an ATM 1483 subinterface take precedence over the values in a VC class assigned to an ATM major interface, but have a lower precedence than the values in a VC class assigned to an individual ATM PVC.

This means that if a VC class is assigned to a PVC configured on the subinterface, the attribute values configured in the VC class assigned to the individual PVC override the attribute values configured in the VC class assigned to the subinterface.

For examples that illustrate how precedence levels affect the assignment of VC classes, see *Precedence Level Examples for Assigning VC Classes* on page 64.

class-int

- Use from Subinterface Configuration mode to assign a previously configured VC class to a static ATM 1483 subinterface.
- You must issue the **exit** command from Subinterface Configuration mode for the VC class association to take effect.
- Example

  ```
  host1(config-subif)#class-int silver-subscriber-class
  host1(config-subif)#exit
  ```

- Use the **no** version to remove the VC class association with the subinterface. Issuing the **no** version causes the router to set the VC attributes to their systemwide default values, or to the values set in the associated VC class with the next highest order of precedence.

Assigning VC Classes to Base Profiles for Bulk-Configured VC Ranges

To assign a VC class to a base profile for a dynamic ATM 1483 subinterface, you can use the **atm class-vc** command from Profile Configuration mode. Issuing this command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this profile.

For more information, see *Configuring ATM 1483 Dynamic Subinterfaces* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Precedence Level Examples for Assigning VC Classes

The examples in this section illustrate how the precedence level rules described in *Precedence Levels* on page 53 affect the assignment of VC classes and PVC attribute values.

For all of these examples, assume that you have issued the following commands to configure a VC class named my-premium-class:

```
host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#encapsulation aal5autoconfig
host1(config-vc-class)#cbr 200
host1(config-vc-class)#oam-pvc manage 60
host1(config-vc-class)#oam ais-rdi 5
host1(config-vc-class)#exit
```

Example 1 and Example 2 illustrate the effect of precedence levels when you assign the VC class my-premium-class to an individual PVC with VCD 200, VPI 0, and VCI 200. Example 3 illustrates how using the **atm pvc** command affects VC class assignment. Finally, Example 4 illustrates how modifications to a VC class affect PVC attributes applied through RADIUS.

Example 1: Explicitly Changing the Service Category

Explicitly specified attribute values take precedence over attribute values specified in a VC class. As a result, the following commands cause the router to use the most recent explicitly specified value, UBR with a PCR of 200 Kbps, as the service category for this PVC instead of the service category specified in my-premium-class, CBR with a PCR of 200 Kbps. The router takes the values for the other attributes from the VC class my-premium-class.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#ubr 200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

The following commands change the service category for the PVC to VBR-RT because this is the most recent explicitly specified value for this attribute. The router takes the values for the other attributes from the VC class my-premium-class, which is still assigned to the PVC.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#vbr-rt 200 150 200
host1(config-subif-vc)#exit
```

The following commands cause the router to retain the VBR-RT service category for the PVC because it is still the most recent explicitly specified value for this attribute. The router takes the values for the other attributes from the VC class my-premium-class.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

Example 2: Changing the Encapsulation Method in the VC Class

The following commands change the value for the encapsulation method in the VC class my-premium-class from **aal5autoconfig** to **aal5snap**. As a result, the router now uses **aal5snap** instead of **aal5autoconfig** as the encapsulation method for the PVCs to which this VC class is assigned.

```
host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#encapsulation aal5snap
host1(config-vc-class)#exit
```

Example 3: Effect of Using the atm pvc Command

The following commands, which attempt to assign the my-premium-class VC class to a PVC originally created with the **atm pvc** command, have no effect. The router interprets all attribute values specified with the **atm pvc** command as explicitly specified values, and therefore takes the values for these attributes from the **atm pvc** command instead of from the VC class. As a result, the router continues to use **aal5mux ip** as the encapsulation method for this PVC instead of the encapsulation method specified in the VC class my-premium-class.

```
host1(config)#interface atm 2/0.300
host1(config-subif)#atm pvc 300 0 300 aal5mux ip
host1(config-subif)#pvc 300 0/300
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
```

Example 4: Overriding RADIUS Values

If RADIUS is configured to provide traffic parameters for PVCs, a more recent, explicitly specified change in the VC class associated with that PVC overrides the PVC values applied through RADIUS.

In the following example, assume that RADIUS has been configured to apply a service category of CBR with a PCR of 400 Kbps to the PVC. Initially, the PVC uses the service category configured in my-premium-class, CBR with a PCR of 200 Kbps. However, when the subscriber logs in through RADIUS, the router applies the RADIUS-configured service category, CBR with a PCR of 400 Kbps.

While the subscriber is still logged in, my-premium-class is modified to change the service category to CBR with a PCR of 600 Kbps. Because this VC class modification results in the most recent, explicitly specified value for the service category, the router now uses CBR with a PCR of 600 Kbps as the service category for the PVC instead of the service category configured through RADIUS.

```
host1(config)#interface atm 2/0.200
host1(config-subif)#pvc 200 0/200
host1(config-subif-vc)#class-vc my-premium-class
host1(config-subif-vc)#exit
! Subscriber logs in through RADIUS, which applies service category of CBR
! with a PCR of 400 Kbps to PVC.
host1(config)#vc-class atm my-premium-class
host1(config-vc-class)#cbr 600
host1(config-vc-class)#exit
! Router now applies service category of CBR with a PCR of 600 Kbps to PVC.
```

Configuring Dynamic ATM 1483 Subinterfaces

As an alternative to the static ATM interface configurations described in this chapter, you can also configure dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces over ATM. Dynamic ATM 1483 subinterfaces can perform autodetection and dynamic creation of the following upper-layer encapsulation types:

- Bridged Ethernet
- IP
- PPP
- PPPoE

For details, see *Configuring ATM 1483 Dynamic Subinterfaces* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Monitoring ATM

This section explains how to set a statistics baseline, display bit rate and packet rate statistics for ATM virtual circuits (VCs), and use the **show** commands to view your ATM configuration and monitor ATM VCs and VPs.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

Setting Statistics Baselines

You can set a statistics baseline for ATM interfaces, ATM virtual circuits, and ATM virtual paths configured on the router.

baseline atm vp interface

- Use to set a statistics baseline for an ATM virtual path (VP) interface.
- 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.
- To set the baseline for an ATM VP, specify the VPI. The numeric range of the VPI depends on the line module capabilities and current configuration.
- To display baseline statistics, use the **delta** keyword with ATM **show** commands.
- Examples

  host1#**baseline atm vp interface atm 12/0 0**
  host1#**baseline atm vp interface atm 5/0/0 1**

- There is no **no** version.

baseline interface atm

- Use to set a statistics baseline for ATM interfaces or a specific virtual circuit.
- 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.
- To set the baseline for a circuit, specify a VCD in the range 1–2147483647.
- To set the baseline on an interface, omit the VCD.
- To display baseline statistics, use the **delta** keyword with ATM **show** commands.
- Examples

  host1#**baseline interface atm 9/1 123**
  host1#**baseline interface atm 5/0/0 123**

- There is no **no** version.

Displaying Interface Rate Statistics for ATM VCs and ATM VPs

You can use the following commands to display bit rate and packet rate statistics over a specified time interval for one or more ATM virtual circuits (VCs) or virtual paths (VPs) configured on the router.

- To monitor the data rate for ATM VCs, use the **monitor atm vc** command.
- To monitor the data rate for ATM VPs, use the **monitor atm vp** command.

To monitor the data rate for ATM VCs and ATM VPs:

1. Log in to the router by using a local console session or a virtual terminal (vty) session (such as a Telnet session).

   While you use the **monitor atm vc** command or the **monitor atm vp** command, you must keep the console or terminal session open. You cannot issue any other commands during the session.

   For information about logging in to the router, see *Accessing the CLI* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

2. Access User Exec mode or Privileged Exec mode.

   For information, see *Accessing Command Modes* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

3. Specify the interface identifier and VCD (for each ATM VC) or VPI (for each ATM VP) that you want to monitor. For information about specifying an ATM interface, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

   host1#**monitor atm vc atm 12/0 1 atm 8/0 1 display-time-of-day**

   host1#**monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day**

By default, the router uses a 5-second time interval between polls to calculate bit rates and packet rates for each specified VC or VP. Optionally, you can use the **load-interval** keyword to specify a nondefault time interval in the range 5–30 seconds (for ATM VCs) or 5–300 seconds (for ATM VPs).

You can also include the optional **display-time-of-day** keyword to show the time of day at which the router gathers statistics for each interval. Displaying the time of day enables you to monitor when a particular VC or VP is underutilized or overutilized.

4. Review the command output.

```
host1#monitor atm vc atm 12/0 1 atm 8/0 1 display-time-of-day
                            Seconds
                            between                                    Time
       Interface         VCD  polls    Input bps/pps   Output bps/pps  (UTC)
----------------------- ----- -------- --------------- --------------- --------
ATM 12/0                    1        0           --/--           --/-- 10:43:11
ATM 8/0                     1        0           --/--           --/-- 10:43:11
ATM 12/0                    1        5     121840/100      121840/100 10:43:16
ATM 8/0                     1        5     121600/100      121600/100 10:43:16
ATM 12/0                    1        5       98008/80        98008/80 10:43:21
ATM 8/0                     1        5       98008/80        98008/80 10:43:21
...

host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day
                            Seconds
                            between                                    Time
       Interface         VPI  polls    Input bps/pps  Output bps/pps  (UTC)
----------------------- ----- -------- --------------- --------------- --------
ATM 12/0                    0        0           --/--           --/-- 09:47:18
ATM 12/0                    1        0           --/--           --/-- 09:47:18
ATM 12/0                    0       15   6635792/6480    6635792/6480 09:47:33
ATM 12/0                    1       15   6635312/6479    6635312/6479 09:47:33
ATM 12/0                    0       15   6635176/6479    6635176/6479 09:47:48
ATM 12/0                    1       15   6634424/6478    6634424/6478 09:47:48
ATM 12/0                    0       15   6635448/6479    6635448/6479 09:48:03
```

The **monitor atm vc** command and **monitor atm vp** command display similar information, except that the **monitor atm vc** command displays the VCD for each interface and the **monitor atm vp** command displays the VPI for each interface.

The router polls the statistics of each VC or VP identified in the command at the specified load interval to calculate and display bit rate and packet rate statistics. The first line of output for each interface always displays 0 (zero) for the number of seconds between polls, and dashes (--/--) in the Input bps/pps and Output bps/pps columns. These values indicate that the router initially takes a baseline for each interface against which to measure subsequent statistics. The router continues to display subsequent lines of output for each interface at the specified load interval until you press Ctrl + c to stop the command.

For a description of the fields in the command display, see **monitor atm vc** and **monitor atm vp** on page 69.

5. If you remove an ATM interface or (for VCs) an ATM 1483 subinterface while you are monitoring one or more VCs or VPs that reside on the deleted interface, press Ctrl + c to stop the **monitor atm vc** command or **monitor atm vp** command, and then restart the command to ensure accurate interface rate statistics are displayed.

   If you leave the **monitor atm vc** command or **monitor atm vp** command running after you remove the interface, the command displays undefined or inaccurate statistics for the VC or VP on the interface that has been removed. The display of undefined or inaccurate statistics can result when you remove the interface by issuing either the **no interface atm** command or **slot erase** command, and can continue even after you recreate the interface with the same VC or VP configuration.

6. When you are finished monitoring, press Ctrl + c to stop the **monitor atm vc** command or **monitor atm vp** command.

   ```
   host1#^C
   ```

monitor atm vc
monitor atm vp

- Use the **monitor atm vc** command to display bit rate and packet rate statistics over a specified time interval for one or more ATM VCs.
- Use the **monitor atm vp** command to display bit rate and packet rate statistics over a specified time interval for one or more ATM VPs.
- You must use either command in a dedicated console or terminal session for the duration of the monitoring session.
- Specify the interface identifier and VCD (for each ATM VC) or VPI (for each ATM VP) that you want to monitor.
- To specify a nondefault time interval in the range 5–30 seconds (for ATM VCs) or 5–300 seconds (for ATM VPs) at which the router calculates bit rate and packet rate statistics, use the optional **load-interval** keyword. The default time interval for either command is 5 seconds.
- To display the time at which the router calculates bit rate and packet rate statistics for the current interval, use the optional **display-time-of-day** keyword.
- To stop either command, press Ctrl + c.
- Field descriptions
  - Interface—Interface identifier for the ATM interface on which the VC or VP resides
  - VCD—Virtual circuit descriptor that identifies the VC (**monitor atm vc** command only)
  - VPI—Virtual path identifier of the PVC (**monitor atm vp** command only)
  - Seconds between polls—Number of seconds at which the router calculates bit rate and packet rate statistics
  - Input bps/pps—Number of bits per second (bps) and packets per second (pps) received on this interface during the specified load interval

  - Output bps/pps—Number of bits per second (bps) and packets per second (pps) transmitted on this interface during the specified load interval
  - Time—Time of day, in hh:mm:ss format, at which the router calculates the bit rate and packet rate statistics for the current interval

- Example 1—Displays bit rate and packet rate statistics over the default (5-second) load interval for a single ATM VC

```
host1#monitor atm vc atm 12/0 100
                              Seconds
                              between
       Interface         VCD   polls    Input bps/pps  Output bps/pps
----------------------- ----- -------- --------------- ---------------
ATM 12/0                  100        0           --/--           --/--
ATM 12/0                  100        5    6631624/6476    6631624/6476
ATM 12/0                  100        5    6630808/6475    6631008/6475
ATM 12/0                  100        5    6632448/6477    6632240/6476
ATM 12/0                  100        5    6629168/6473    6629168/6473
ATM 12/0                  100        5    6631008/6475    6631216/6475
host1#^C
```

- Example 2—Displays bit rate and packet rate statistics over the default (5-second) load interval for two ATM VCs, with the time of day that the statistics were calculated

```
host1#monitor atm vc atm 12/0 100 atm 12/0 200 display-time-of-day
                              Seconds
                              between                                  Time
       Interface         VCD   polls    Input bps/pps  Output bps/pps   (UTC)
----------------------- ----- -------- --------------- --------------- --------
ATM 12/0                  100        0           --/--           --/-- 17:33:06
ATM 12/0                  200        0           --/--           --/-- 17:33:06
ATM 12/0                  100        5    6635104/6479    6635104/6479 17:33:11
ATM 12/0                  200        5    6633264/6477    6633472/6478 17:33:11
ATM 12/0                  100        5    6632856/6477    6632856/6477 17:33:16
ATM 12/0                  200        5    6633264/6477    6633056/6477 17:33:16
host1#^C
```

- Example 3—Displays bit rate and packet rate statistics over a 10-second load interval for two ATM VPs

```
host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 10
                              Seconds
                              between
       Interface         VPI   polls    Input bps/pps  Output bps/pps
----------------------- ----- -------- --------------- ---------------
ATM 12/0                    0        0           --/--           --/--
ATM 12/0                    1        0           --/--           --/--
ATM 12/0                    0       10    6635312/6479    6635312/6479
ATM 12/0                    1       10    6634288/6478    6634288/6478
ATM 12/0                    0       10    6637664/6482    6637664/6482
ATM 12/0                    1       10    6637872/6482    6637872/6482
host1#^C
```

- Example 4—Displays bit rate and packet rate statistics over a 15-second load interval for two ATM VPs, with the time of day that the statistics were calculated

```
host1#monitor atm vp atm 12/0 0 atm 12/0 1 load-interval 15 display-time-of-day
                            Seconds
                            between                                      Time
       Interface         VPI   polls    Input bps/pps  Output bps/pps   (UTC)
----------------------- ----- -------- --------------- --------------- --------
ATM 12/0                    0        0           --/--           --/-- 17:36:19
ATM 12/0                    1        0           --/--           --/-- 17:36:19
ATM 12/0                    0       15    6634352/6478    6634352/6478 17:36:34
ATM 12/0                    1       15    6633608/6478    6633608/6478 17:36:34
ATM 12/0                    0       15    6635176/6479    6635176/6479 17:36:49
ATM 12/0                    1       15    6635040/6479    6635040/6479 17:36:49
host1#^C
```

- There is no **no** version.

Using ATM show Commands

Use the **show** commands described in this section to display information about your ATM configuration and monitor ATM interfaces.

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 *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

show atm aal5 interface

- Use to display information about a configured ATM AAL5 interface.
- Field descriptions
  - AAL5 Interface operational status—Operational status of the AAL5 interface: up, down, lowerlayerDown
  - time since last status change—Time since last reported change to the AAL5 interface operational status
  - SNMP trap link-status—Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
  - Auto configure ATM 1483 status—Setting of the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface:
    - enabled—Autodetection of the ATM 1483 dynamic encapsulation type is enabled on the ATM AAL5 interface
    - disabled—Autodetection of the ATM 1483 dynamic encapsulation type is not currently enabled on the ATM AAL5 interface
  - InPackets—Number of packets received on this interface
  - InBytes—Number of bytes received on this interface
  - OutPackets—Number of packets transmitted on this interface
  - OutBytes—Number of bytes transmitted on this interface
  - InErrors—Number of incoming errors received on this interface
  - OutErrors—Number of outgoing errors on this interface

- InPacketDiscards—Number of incoming packets discarded on this interface
- OutDiscards—Number of outgoing packets discarded on this interface

- Example

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46

SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

show atm atm1483

- Use to display whether or not the router is set up to export ATM 1483 subinterface descriptions to the line module.
- Example

```
host1#show atm atm1483
ATM1483 IF Descriptions exported
```

show atm interface
show interfaces atm

- Use to display configuration and state information and statistics about a specific ATM interface, or to display a brief description of all ATM interfaces configured in the router.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module; on the OC3-2 GE APS I/O module, you can specify ATM interfaces only in ports 0 and 1; port 2 is reserved for a Gigabit Ethernet interface
- To specify an ATM interface for the E120 router and the E320 router, use the *slot*/*adapter*/*port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA

- To display the status and number of configured VCs for all ATM interfaces configured in the router, use the **brief** keyword.
- Field descriptions
  - ATM Interface status—State of the physical interface: up, down
  - line protocol—State of the ILMI protocol: disabled, up, down
  - AAL5 operational status—Operational status of the ATM AAL5 interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the AAL5 operational status
  - ATM operational status—Operational status of the ATM interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the ATM operational status
  - UNI version—UNI version: 3.0, 3.1, 4.0
  - Maximum VCs—Maximum number of virtual circuits supported on this interface
  - Current VCs—Current number of virtual circuits configured
  - ILMI VPI/VCI—Number of VPI and VCI configured for ILMI (displayed only when ILMI is configured on the interface)
  - VCD—Number of VCD (displayed only when ILMI is configured on the interface)
  - ILMI keepalive—State and status of the ILMI (displayed only when ILMI is configured on the interface)
  - Max VCI per VPI—Maximum number of virtual circuits on each virtual path
  - CAC admin state—Enabled, disabled
  - Subscription bandwidth—Maximum allowable bandwidth on the port (displayed only when CAC is enabled)
  - UBR weight—Configured bandwidth for UBR and UBR-PCR connections (displayed only when CAC is enabled)
  - Available bandwidth—Bandwidth currently available on the port (displayed only when CAC is enabled)
  - SNMP trap link-status—Enabled, disabled
  - OAM cell receive status—Whether the ATM interface processes or flushes OAM cells: enabled, disabled
  - OAM cell filter—Whether the interface flushes all OAM cells or flushes only AIS and RDI alarm cells (displayed only when OAM cell receive status is enabled)
  - atm oam loopback-location—Loopback location ID for this interface
  - Interface Alias—Text description or alias if configured for the interface
  - Assigned VC Class—Name of the VC class assigned to this interface, if configured
  - InPackets—Number of packets received on this interface

- InBytes—Number of bytes received on this interface
- InCells—Number of cells received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- OutCells—Number of cells transmitted on this interface
- InErrors—Number of incoming errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- InByteDiscards—Number of incoming bytes discarded on this interface
- InCellErrors—Increments when a T3 or an E3 ATM interface receives cells for a VPI or VCI that is not configured on that interface

- Field descriptions specific to the applicable physical interface. In Example 1, the output contains the following physical interface fields:
  - SONET path operational status—State of the SONET path interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the SONET path operational status
  - SONET operational status—State of SONET interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the SONET operational status
  - PHY Type—Physical port type on which this interface is running
  - Framing—Framing type of the physical interface
  - TX clocking—Clocking type for the physical interface
  - Loopback—Loopback status for the physical interface: enabled, disabled
  - Receive FIFO Overruns—Number of times received FIFO was overrun
  - qos-mode-port—Status of SAR backpressure: enabled, disabled
  - queue—Hardware packet queue associated with the specified traffic class and interface
  - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
  - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets 0, Bytes 0—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets 0, Bytes 0—Number of exceeded packets and bytes that were dropped
  - Interface—ATM interface identifier
  - Status—Status of the ATM interface: up, down, lowerLayerDown
  - Configured VCs—Number of VCs configured on the interface

- Example 1—Displays information about a specific interface

```
host1#show atm interface atm 2/0
ATM Interface 2/0 is down, line protocol is down

AAL5 operational status:       lowerLayerDown
      time since last status change: 22:08:21
ATM operational status:        down
      time since last status change: 22:02:11
SONET path operational status: lowerLayerDown
      time since last status change: 1 day, 0 hours
SONET operational status:      down
      time since last status change: 1 day, 0 hours
UNI version: 3.0, Maximum VCs: 4096
Current VCs: 1
ILMI VPI/VCI: 17/23, VCD 26, ILMI keepalive: disabled
Max VCI per VPI: 32768
CAC admin state: enabled
Subscription bandwidth: 3000000 kbps
UBR weight: 3000 kbps
Available bandwidth: 2992000 kbps
SNMP trap link-status: enabled
OAM cell receive status: enabled
OAM cell filter : all cells
atm oam loopback-location 0XFFFFFFFF
Interface Alias: ATM interface slot #2 port 0
Assigned VC class       : dsl-subscriber-class

PHY Type: oc3, Framing: sonet, TX clocking: line
Loopback: none, Receive FIFO Overruns: 0

5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec

InPackets:        0
InBytes:          0
InCells:          0
OutPackets:       0
OutBytes:         0
OutCells:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
InByteDiscards:   0
InCellErrors:     0
qos-mode-port disabled

queue 0: traffic class control, bound to ATM2/0
  Forwarded packets 643, Bytes 36008
  Dropped committed packets 0, Bytes 0
  Dropped conformed packets 0, Bytes 0
  Dropped exceeded packets 0, Bytes 0
```

- Example 2—Shows a summary of all ATM interfaces

```
host1#show atm interface brief
                                Configured
Interface       Status             VCs
---------   --------------   ----------
ATM 2/0     up                        2
ATM 2/1     up                        3
ATM 2/2     lowerLayerDown            4
ATM 2/3     down                      5
ATM 4/0     up                        2
ATM 6/0     lowerLayerDown            2
```

show atm map

- Use to display the list of all configured ATM static maps to remote hosts on an ATM network.
- Field descriptions
  - Map list—Name of map list and method used to enter the map list. PERMANENT indicates that the map entry was configured; it was not entered automatically by a process.
  - protocol address maps to VCx—Name of protocol, the protocol address, and the VCD that the address is mapped to (for ATM VCs configured with the **atm pvc** command).
    - VC—Number of the virtual circuit
    - broadcast—Indicates pseudo-broadcasting
    - connection up—Indicates a point-to-point virtual circuit
- Example 1

```
host1#show atm map
Map list my-map : PERMANENT
 ip 192.168.2.10 maps to VC 10 atm 2/0
 ip 192.168.2.20 maps to VC 11 atm 2/0    broadcast
 ip 192.168.2.30 maps to VC 12 atm 2/0
Map list other-map : PERMANENT
 ip 192.10.2.10 maps to VC 100 atm 2/1
 ip 192.10.2.20 maps to VC 101 atm 2/1
 ip 192.10.2.30 maps to VC 102 atm 2/1    broadcast
```

- Example 2

```
host1#show atm map brief
Map list my-map : PERMANENT
Map list other-map : PERMANENT
```

- Example 3

```
host1#show atm map my-map
Map list my-map : PERMANENT
 ip 192.168.2.10 maps to VC 10 atm 2/0
 ip 192.168.2.20 maps to VC 11 atm 2/0    broadcast
 ip 192.168.2.30 maps to VC 12 atm 2/0
```

show atm oam

- Use to display F4 OAM statistics for an ATM interface.
- You must specify a VPI value in addition to the required ATM interface specifier.
- You can use the following keywords.
  - **segment**—Displays information about segment loopbacks
  - **end-to-end**—Displays information about end-to-end loopbacks
- To see F4 OAM circuits that are open, use the **show atm vc** command.
- Field descriptions
  - Sending End To End Loopback Cells—Enabled, disabled
  - Frequency—Frequency configured on this circuit
  - End To End OAM CC verification—Whether end-to-end CC verification is enabled or disabled
  - OAM CC Type—Whether the circuit is a sink or a source, or both a sink and a source
  - OAM Current CC state
    - Ready—OAM CC is not enabled
    - Active—OAM CC cell flow is running
    - Activation Failed—OAM CC activation failed
    - Wait Activate—Waiting for interface to come up before the software sends the activation request
    - Wait Activation Confirmation—Waiting for activation confirmation from the peer
    - Wait DeActivate—Waiting for interface to come up before the software sends the deactivation request
    - Wait DeActivation Confirmation—Waiting for deactivation confirmation from the peer
  - Segment OAM CC verification—Whether segment CC verification is enabled or disabled
  - VP State—State of the VP: up, down
  - VP End To End Oam State
    - not managed—Circuit is in normal OAM state; no OAM fault conditions
    - AIS—Circuit is in AIS state
    - RDI—Circuit is in RDI state
  - VP Segment Oam State
    - not managed—Circuit is in normal OAM state; no OAM fault conditions
    - AIS—Circuit is in AIS state
    - RDI—Circuit is in RDI state
  - InOamF4Cells—Number of F4 OAM cells received
  - InOamF4CellsDropped—Number of incoming F4 OAM cells that were dropped

- InOamF4EndLoopbackCells—Total number of F4 end-to-end loopback cells received on this interface, which is the sum of the following counts:
  - InOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands received
  - InOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses received
- InOamF4SegLoopbackCells—Total number of F4 segment loopback cells received on this interface, which is the sum of the following counts:
  - InOamF4SegLoopbackCommands—Number of F4 segment loopback commands received
  - InOamF4SegLoopbackResponses—Number of F4 segment loopback responses received
- InOamF4EndAisCells—Number of F4 end-to-end AIS cells received
- InOamF4SegAisCells—Number of F4 segment AIS cells received
- InOamF4EndRdiCells—Number of F4 end-to-end RDI cells received
- InOamF4SegRdiCells—Number of F4 segment RDI cells received
- InOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells received
- InOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells received
- InOamF4EndCCCells—Number of F4 end-to-end CC cells received
- InOamF4SegCCCells—Number of F4 segment CC cells received
- OutOamF4Cells—Number of F4 OAM cells sent
- OutOamF4EndLoopbackCells—Total number of F4 end-to-end loopback cells sent on this interface, which is the sum of the following counts:
  - OutOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands sent
  - OutOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses sent
- OutOamF4SegLoopbackCells—Total number of F4 segment loopback cells sent on this interface, which is the sum of the following counts:
  - OutOamF4SegLoopbackCommands—Number of F4 segment loopback commands sent
  - OutOamF4SegLoopbackResponses—Number of F4 segment loopback responses sent
- OutOamF4EndRdiCells—Number of end-to-end RDI cells sent
- OutOAM F4SegRdiCells—Number of segment RDI cells sent
- OutOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells sent
- OutOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells sent

- OutOamF4EndCCCells—Number of F4 end-to-end CC cells sent
- OutOamF4SegCCCells—Number of F4 segment CC cells sent

- Example 1

```
host1#show atm oam 2/1 0
Sending End To End Loopback Cells is Enabled: Frequency = 20 secs
End To End OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state : Ready
Segment OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state : Ready
VP State                          :down
VP End To End Oam State           :not managed
VP Segment Oam State              :not managed
InOamF4Cells                      :0
InOamF4CellsDropped               :0
InOamF4EndLoopbackCells           :0
  InOamF4EndLoopbackCommands          :0
  InOamF4EndLoopbackResponses         :0
InOamF4SegLoopbackCells           :0
  InOamF4SegLoopbackCommands          :0
  InOamF4SegLoopbackResponses         :0
InOamF4EndAisCells                :0
InOamF4SegAisCells                :0
InOamF4EndRdiCells                :0
InOamF4SegRdiCells                :0
InOamF4EndCCActDeActCells         :0
InOamF4SegCCActDeActCells         :0
InOamF4EndCCCells                 :0
InOamF4SegCCCells                 :0
OutOamF4Cells                     :0
OutOamF4EndLoopbackCells          :0
  OutOamF4EndLoopbackCommands         :0
  OutOamF4EndLoopbackResponses        :0
OutOamF4SegLoopbackCells          :0
  OutOamF4SegLoopbackCommands         :0
  OutOamF4SegLoopbackResponses        :0
OutOamF4EndRdiCells               :0
OutOamF4SegRdiCells               :0
OutOamF4EndCCActDeActCells        :0
OutOamF4SegCCActDeActCells        :0
OutOamF4EndCCCells                :0
OutOamF4SegCCCells                :0
Time since last status change     :00:00:33
```

- Example 2

```
host1#show atm oam 2/1 0 segment
Segment OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state: Ready
VP State                         :down
VP Oam State                     :not managed
InOamF4SegmentCells               :0
InOamF4SegmentCellsDropped        :0
InOamF4SegLoopbackCells           :0
  InOamF4SegLoopbackCommands              :0
  InOamF4SegLoopbackResponses             :0
InOamF4SegCCActDeActCells         :0
InOamF4SegCCCells                 :0
OutOamF4SegmentCells              :0
```

```
OutOamF4SegLoopbackCells         :0
  OutOamF4SegLoopbackCommands             :0
  OutOamF4SegLoopbackResponses            :0
OutOamF4SegRdiCells              :0
OutOamF4SegCCActDeActCells       :0
OutOamF4SegCCCells               :0
Time since last status change    :00:00:53
```

- Example 3

```
host1#show atm oam 2/1 0 end-to-end
Sending End To End Loopback Cells Disabled:
End To End OAM CC verification enabled
OAM CC Type : CC Sink End Point
OAM Current CC state: Ready
VP State                          :down
VP Oam State                      :not managed
InOamF4EndCells                  :0
InOamF4EndCellsDropped           :0
InOamF4EndLoopbackCells          :0
  InOamF4EndLoopbackCommands              :0
  InOamF4EndLoopbackResponses             :0
InOamF4EndAisCells :0
InOamF4EndRdiCells :0
InOamF4EndCCActDeActCells        :0
InOamF4EndCCCells                :0
OutOamF4EndCells                 :0
OutOamF4EndLoopbackCells         :0
  OutOamF4EndLoopbackCommands             :0
  OutOamF4EndLoopbackResponses            :0
OutOamF4EndRdiCells              :0
OutOamF4EndCCActDeActCells       :0
OutOamF4EndCCCells               :0
Time since last status change    :00:00:53
```

show atm ping

- Use to show all existing ping entries, both completed and outstanding. Remember that ping statistics are overwritten when a new ping is issued on the circuit.
- You can specify the following options to show ping for entries for a specific interface, VPI, or VCI.
  - *interfaceSpecifier*—Shows ping entries for this interface
  - *vpi*—Shows details of the last **ping atm** command on this VPI
  - *vci*—Shows details of the last **ping atm** command on this VCI
- Field descriptions
  - Interface—Interface number
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - CellCount—OAM loopback cell count configured on the interface
  - TimeOut—Timeout configured on the interface
  - SentCellCount—Number of loopback cells sent
  - RespCount—Number of loopback response cells received

- Status—Status of the ping
- Ping Cell Count—Cell count configured on the circuit
- Ping Time Out—Timeout, in seconds, configured on the circuit
- No Of Cells Sent—Number of ping cells sent on this circuit
- No Of Response Received—Number of ping responses received on this circuit
- Success Rate—Percentage of successful responses received for pings sent
- round-trip min/max/avg—Minimum, maximum, and average time in milliseconds that it took to receive responses to ping messages sent
- Ping Status—Results of the ping operation
  - Ping Completed—Number of ping requests in the cell count were sent
  - Ping in Progress—Ping is in operation
  - Ping Not Started—Ping operation is not started; you may see this via SNMP
  - Ping Stopped—Ping operation was manually stopped
  - Ping Stopped OAM Down—**atm oam flush** command was issued when ping was enabled
  - ATM Interface Down—Ping operation is stopped as a result of interface down operational status
- OAM Flow Type—Segment, End-to-end

- Example 1—Displays all entries in the router

```
host1#show atm ping
Interface VPI VCI CellCount TimeOut  SentCellCount  RespCount    Status
--------- --- --- --------- -------  -------------  ----------   -------
ATM 2/1   0   100         5       5              5           5   Ping Completed
ATM 2/1   0   200         5       5              5           5   Ping Completed
ATM 2/2   0   100         5       5              5           5   Ping Completed
ATM 2/2   0   200         5       5              5           5   Ping Completed
% Found 4  Entries in the system
```

- Example 2—Displays entries on an interface

```
host1#show atm ping 2/1
Interface VPI VCI  CellCount TimeOut  SentCellCount  RespCount    Status
--------- --- ---  --------- -------  -------------  ----------   -------
ATM 2/1   0   100          5       5              5           5   Ping Completed
ATM 2/1   0   200          5       5              5           5   Ping Completed
% Found 2  Entries in this Interface
```

- Example 3—Displays entries on a circuit

```
host1#show atm ping atm 2/1 0 100
Ping Cell Count            :5
Ping Time Out              :5secs
No Of Cells Sent           :5
No Of Response Received    :5
Success Rate               :100%
round-trip min/max/avg     :0/10/2 ms
Ping Status                :Completed
Oam Flow Type              :Segment
```

show atm subinterface

- Use to display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface.
- To specify an ATM subinterface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM subinterface for the E120 router and the E320 router, use the *slot*/*adapter*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface, use the **summary** keyword.
- To display status information only for ATM subinterfaces with a specific operating status, use the **status** keyword with one of the following status values. (See the Status field description for an explanation of these values.)
  - dormant
  - dormantLockout
  - down
  - lowerLayerDown
  - notPresent
  - up

- To display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values, use the **atm** *slot/port/vpi/vci* format (for ERX-7xx models, ERX-14xx models, and ERX-310 routers) or the *slot/adapter/port/vpi/vci* format (for E120 routers and E320 routers) to identify the ATM subinterface (Example 5).

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 routers and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

- For more information, see *Creating a Basic Configuration* on page 20.
- Field descriptions
  - Interface—Interface identifier
  - ATM-Prot—One of the following ATM protocol types:
    - RFC-1483—Multiprotocol encapsulation over AAL5
    - NBMA—Nonbroadcast multiaccess interface
    - ATM/MPLS—Local ATM passthrough interface
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual circuit (or channel) identifier
  - Circuit Type—Type of circuit: PVC
  - Encap—Administered encapsulation method based on what was configured with the **atm pvc** command
  - MTU—Maximum transmission unit size for the interface
  - Status—One of the following ATM 1483 subinterface states:
    - absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled.
    - dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface.

  - dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout.
  - down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured.
  - lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down.
  - up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.
- Interface Type—Type of ATM 1483 subinterface: dynamic or static
- Auto configure status—Setting of the autoconfiguration feature
  - dynamic—Autodetection is on; the router automatically detects the next upper interface
  - static—Autodetection is off
- Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE
- Detected 1483 encapsulation—If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
- Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
- Interface types in lockout—Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none
- Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Next—Lockout time for the router to use for the next lockout event, in seconds
- Assigned profile—For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name
- Interface Alias—Text description or alias if configured for the subinterface

- Subscriber info—Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
- Assigned VC Class—Name of the VC class assigned to this subinterface, if configure
- SNMP trap link-status—Trap link status: enabled or disabled
- Advisory receive speed—Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E-series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- InPacketsUnknownProtocol—Number of incoming packets with an unknown protocol type
- OutDiscards—Number of outgoing packets discarded on this interface

- Example 1—Displays the current state of all ATM subinterfaces

```
host1#show atm subinterface
                                    Circuit                              Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status            Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102   0 102 PVC     AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103   0 103 PVC     AUTO  9180 dormant        Static
3 interface(s) found
```

- Example 2—Displays summary information for all ATM subinterfaces shown in Example 1

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

- Example 3—Displays status information about the current state of all ATM subinterfaces in the dormantLockout state

```
host1#show atm subinterface status dormantLockout
                                    Circuit                              Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status            Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
1 interface(s) found
```

- Example 4—Displays the current state of a specific ATM subinterface

```
host1#show atm subinterface atm 2/0.101
                                    Circuit                           Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP
Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------ --- ---- ------- ------- ----
IP                               1   30      16       7   30
BridgedEnet                    900 3600       0       0  900
PPP                              1  300       0       0    1
PPPoE                            1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

Interface Alias: atm20101

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

Assigned VC class             : premium-subscriber-class
SNMP trap link-status: disabled

InPackets:                0
InBytes:                  1904
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         14
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

- Example 5—Displays the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values

```
host1#show atm subinterface atm 0/0/0/101
                                    Circuit                   Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none
```

```
Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                            1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

Assigned VC class             : dsl-subscriber-class
SNMP trap link-status: disabled

Advisory receive speed: 2000

InPackets:                5119
InBytes:                  358672
OutPackets:               5107
OutBytes:                 357510
InErrors:                 0
OutErrors:                0
InPacketDiscards:         3
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

show atm vc

- Use to display a summary of all configured ATM virtual circuits (VCs) and reserved VC ranges.
- You can specify one or more of the following keywords individually or in combination:
  - **vpi**—Displays VCs on a specific VPI
  - **category**—Displays VCs that have a specific service category
  - **status**—Displays VCs with a certain status
- You can also specify the **reserved** keyword with no other keywords to display only a summary of all reserved VC ranges on the router. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces and by MPLS.
- Field descriptions
  - Interface—Interface number
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - VCD—Virtual circuit descriptor
  - Type—Type of circuit: PVC
  - Encap—Encapsulation method: AUTO, AAL5, AAL0, MUX, SNAP, ILMI, F4-OAM
  - Category—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
  - Rx/Tx Peak—Peak rate, in Kbps
  - Rx/Tx Avg—Average rate, in Kbps

- Rx/Tx Burst—Maximum number of cells that can be burst at the peak cell rate
- Status—State of the virtual circuit: Up or Down
- Start VPI—Starting virtual path identifier (inclusive) of the reserved VC range
- Start VCI—Starting virtual circuit identifier (inclusive) of the reserved VC range
- End VPI—Ending virtual path identifier (inclusive) of the reserved VC range
- End VCI—Ending virtual circuit identifier (inclusive) of the reserved VC range

- Example 1—Displays all VCs and reserved VC ranges on the router

```
host1#show atm vc
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- Example 2—Displays VCs with a VPI of zero (0)

```
host1#show atm vc vpi 0
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
2 circuit(s) found that match filter criteria
```

- Example 3—Displays VCs with a VPI of 1 (one) that are assigned the service category UBR

```
host1#show atm vc vpi 1 category ubr
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
2 circuit(s) found that match filter criteria
```

- Example 4—Displays VCs with a VPI of 0 (zero) and a service category of CBR that have a status of up

```
host1#show atm vc vpi 0 category cbr status up
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
1 circuit(s) found that match the filter criteria
```

- Example 5—Displays all reserved VC ranges on the router

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

show atm vc atm

- Use to display information about a specific VC.
- To specify the circuit to display, do one of the following:
  - Enter the VCD.
  - Use the **vpi-vci** keyword and enter the VPI and VCI.
  - Enter the description configured for the ATM 1483 subinterface (with the **atm atm1483 description** command) on which the VC resides.
- Field descriptions
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - Encap—Encapsulation method
  - Service Type—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
  - Inverse ARP enable—Whether or not Inverse ARP is enabled: yes, no
  - Assigned VC class—Name of the VC class assigned to this VC, if configured
  - InPackets—Number of packets received on this circuit
  - InBytes—Number of bytes received on this circuit
  - InCells—Number of ATM cells received on this circuit
  - OutPackets—Number of packets transmitted on this circuit
  - OutBytes—Number of bytes transmitted on this circuit
  - OutCells—Number of ATM cells transmitted on this circuit
  - InErrors—Number of errors received on this circuit
  - OutErrors—Number of outgoing errors on this circuit

- InPacketDiscards—Number of incoming packets discarded on this circuit
- InPacketUnknownProtocol—Number of incoming packets with an unknown protocol type
- InByteDiscards—Number of incoming bytes discarded on this circuit
- CrcErrors—Number of CRC errors detected on this circuit
- SAR time-outs—Number of segmentation and reassembly (SAR) timeouts reached on this circuit
- Over-sized SDUs—Number of oversized service data units (SDUs) received on this circuit
- Alarm drop count—Number of successive alarm cells that the router receives before reporting that the PVC is down
- Alarm clear timeout—Number of seconds that the router waits before reporting that the PVC is up after the PVC stops receiving alarm cells
- OAM VC verification—Whether OAM verification is enabled or disabled
- OAM loopback cell status:
  - disabled—VC integrity disabled for VC
  - sent—OAM loopback cell sent; waiting for response
  - received—OAM loopback cell response received
  - failed—OAM loopback reply not received within frequency period, or reply contained a bad correlation tag
- OAM VC status:
  - AIS—VC is in AIS state
  - RDI—VC is in RDI state
  - Down Retry—OAM loopback failed; using retry frequency to verify that the VC is really down
  - Down—OAM loopback failed after Down Retry verification
  - Up Retry—OAM loopback successful; using retry frequency to verify that the VC is really up
  - Up—OAM loopback successful after Up Retry verification
  - Not Managed—VC integrity is not enabled; for more information about this status value, see *Automatic Disabling of F5 OAM Services* on page 18
- OAM loopback frequency—Frequency with which OAM loopback cells are transmitted (when enabled), in seconds
- OAM up retry count—Number of consecutive successfully looped OAM cells required to mark the VC as Up
- OAM down retry count—Number of consecutive unsuccessfully looped OAM cells required to mark the VC as Down
- OAM loopback retry frequency—Frequency with which OAM cells are transmitted in verification mode, in seconds
- OAM CC verification—Whether CC verification is enabled or disabled

- OAM CC Type—Whether the VC is a sink or a source, or both sink and source end point
- OAM CC Flow Type—End-to-end or segment
- OAM Current CC state
  - Ready—OAM CC is not enabled
  - Active—OAM CC cell flow is running
  - Activation Failed—OAM CC activation failed
  - Wait Activate—Waiting for interface to come up before the software sends the activation request
  - Wait Activation Confirmation—Waiting for activation confirmation from the peer
  - Wait DeActivate—Waiting for interface to come up before the software sends the deactivation request
  - Wait DeActivation Confirmation—Waiting for deactivation confirmation from the peer
- InOamF5Cells—Number of F5 OAM cells received on this circuit
- InOamCellDiscards—Number of received OAM cells that were dropped; dropped cells include unsupported and invalid F5 cells. The InOamCellDiscards counter is not incremented after an OAM flush is performed with the **atm oam flush** command. For more information about the InOamCellDiscards counter, see *Rate Limiting for F5 OAM Cells* on page 18.
- InF5EndLoopCells—Total number of F5 end-to-end loopback cells received on this circuit, which is the sum of the following counts:
  - InF5EndLoopCommands—Number of F5 end-to-end loopback commands received
  - InF5EndLoopResponses—Number of F5 end-to-end loopback responses received
- InF5SegLoopCells—Total number of F5 segment loopback cells received on this circuit, which is the sum of the following counts:
  - InF5SegLoopCommands—Number of F5 segment loopback commands received
  - InF5SegLoopResponses—Number of F5 segment loopback responses received
- InF5EndAisCells—Number of F5 end-to-end AIS cells received on this circuit
- InF5SegAisCells—Number of F5 segment AIS cells received on this circuit
- InF5EndRdiCells—Number of F5 end-to-end RDI cells received on this circuit
- InF5SegRdiCells—Number of F5 segment RDI cells received on this circuit
- InF5EndCCActDeActCells—Number of F5 end-to-end activation and deactivation CC cells received on this circuit

- InF5SegCCActDeActCells—Number of F5 segment activation and deactivation CC cells received on this circuit
- InF5EndCCCells—Number of F5 end-to-end CC cells received on this circuit
- InF5SegCCCells—Number of F5 segment CC cells received on this circuit
- OutOamF5Cells—Number of F5 OAM cells transmitted on this circuit
- OutF5EndLoopCells—Total number of F5 end-to-end loopback cells transmitted on this circuit, which is the sum of the following counts:
  - OutF5EndLoopCommands—Number of F5 end-to-end loopback commands transmitted
  - OutF5EndLoopResponses—Number of F5 end-to-end loopback responses transmitted
- OutF5SegLoopCells—Total number of F5 segment loopback cells transmitted on this circuit, which is the sum of the following counts:
  - OutF5SegLoopCommands—Number of F5 segment loopback commands transmitted
  - OutF5SegLoopResponses—Number of F5 segment loopback responses transmitted
- OutF5EndRdiCells—Number of F5 end-to-end RDI cells transmitted on this circuit
- OutF5SegRdiCells—Number of F5 segment RDI cells transmitted on this circuit
- OutF5EndCCActDeActCells—Number of F5 end-to-end activation and deactivation CC cells transmitted on this circuit
- OutF5SegCCActDeActCells—Number of F5 segment activation and deactivation CC cells transmitted on this circuit
- OutF5EndCCCells—Number of F5 end-to-end CC cells transmitted on this circuit
- OutF5SegCCCells—Number of F5 segment CC cells transmitted on this circuit
- Circuit is Up/Down—Status of the circuit and time since the status of the circuit last changed

- Example 1—Displays statistics for the VC with a VPI of 46 and a VCI of 47

```
host1#show atm vc atm 2/0 vpi-vci 46 47
ATM 2/0.1.1: VCD: 45, VPI: 46, VCI: 47, Encap: AAL5-AUTO
Service Type: Ubr
Inverse ARP enable:No
Assigned VC class :premium-subscriber-class
InPackets:         0
InBytes:           0
InCells:           0
OutPackets:        0
OutBytes:          0
OutCells:          0
InErrors:          0
OutErrors:         0
InPacketDiscards:  0
InPacketUnknownProtocol: 0
InByteDiscards:    0
```

```
CrcErrors:         0
SAR time-outs:     0
Over-sized SDUs:   0
Alarm drop count:  1
Alarm clear timeout:3
OAM VC verification: enabled
OAM loopback cell status: sent
OAM VC status: up
OAM loopback frequency: 10 second interval
OAM up retry count: 3, OAM down retry count: 5
OAM loopback retry frequency: 1 second interval
OAM CC verification: disabled
InOamF5Cells:      258
InOamCellDiscards: 12598
InF5EndLoopCells:  258
  InF5EndLoopCommands:    50
  InF5EndLoopResponses:   208
InF5SegLoopCells:  46
  InF5SegLoopCommands:    17
  InF5SegLoopResponses:   29
InF5EndAisCells:        49
InF5SegAisCells:        0
InF5EndRdiCells:        0
InF5SegRdiCells:        0
InF5EndCCActDeActCells: 0
InF5SegCCActDeActCells: 0
InF5EndCCCells:         0
InF5SegCCCells:         0
OutOamF5Cells:     258
OutF5EndLoopCells: 258
  OutF5EndLoopCommands:    208
  OutFEndLoopResponses:    50
OutF5SegLoopCells: 48
  OutF5SegLoopCommands:    19
  OutF5SegLoopResponses:   29
OutF5EndRdiCells:       50
OutF5SegRdiCells:       0
OutF5EndCCActDeActCells:1
OutF5SegCCActDeActCells:0
OutF5EndCCCells:        1
OutF5SegCCCells:        0

Circuit is Up, time since last change: 5 days, 23 hours
```

- Example 2—Displays statistics for the VC that resides on the ATM 1483 subinterface configured with the specified description (myAtm301)

```
host1#show atm vc myAtm301
ATM3/0.1: VCD: 10, VPI: 5, VCI: 100, Encap: SNAP
Service Type: Ubr
Assigned VC class         :dsl-subscriber-class
InPackets:                 0
InBytes:                   0
InCells:                   0
OutPackets:                0
OutBytes:                  0
OutCells:                  0
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          0
InPacketUnknownProtocol:   0
InByteDiscards:            0
CrcErrors:                 0
```

```
SAR time-outs:             0
Over-sized SDUs:           0
Alarm drop count:          1
Alarm clear timeout:       3
OAM VC verification:       disabled
OAM VC status:             not managed
OAM CC verification:       disabled
InOamF5Cells:              0
InOamCellDiscards:         384723
InF5EndLoopCells:          0
  InF5EndLoopCommands:     0
  InF5EndLoopResponses:    0
InF5SegLoopCells:          0
  InF5SegLoopCommands:     0
  InF5SegLoopResponses:    0
InF5EndAisCells:           0
InF5SegAisCells:           0
InF5EndRdiCells:           0
InF5SegRdiCells:           0
InF5EndCCActDeActCells:    0
InF5SegCCActDeActCells:    0
InF5EndCCCells:            0
InF5SegCCCells:            0
OutOamF5Cells:             0
OutF5EndLoopCells:         0
  OutF5EndLoopCommands:    0
  OutFEndLoopResponses:    0
OutF5SegLoopCells:         0
  OutF5SegLoopCommands:    0
  OutF5SegLoopResponses:   0
OutF5EndRdiCells:          0
OutF5SegRdiCells:          0
OutF5EndCCActDeActCells:   0
OutF5SegCCActDeActCells:   0
OutF5EndCCCells:           0
OutF5SegCCCells:           0

Circuit is DOWN, time since last change: 02:25:52
```

show atm vc-class

- Use to display information about the VC classes configured on the router.
- To display only the names of all VC classes configured on the router, use the command with no keywords.
- To display detailed configuration information about a particular VC class, specify the name of the VC class.
- To display the settings for parameters in the VC class that are configured with default values, use the **include-defaults** keyword.
- Field descriptions
  - Encapsulation Type—Encapsulation method configured in the VC class: AUTO, AAL5, AAL0, MUX, SNAP
  - Service Category—Service category configured in the VC class: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
  - Peak Cell Rate—Peak cell rate (PCR), in Kbps, configured for the service category

- OAM VC Integrity—Status of F5 OAM VC integrity features on the PVC: enabled or disabled
- OAM VC Integrity loop-back timer—Number of seconds the router waits between the transmission of loopback cells during normal operation
- OAM VC Integrity Up Retry Count—Number of successive loopback cell responses that the router receives before reporting that a PVC is up
- OAM VC Integrity Down Retry Count—Number of successive loopback cell responses that the router misses before reporting that a PVC is down
- OAM VC Integrity Retry Frequency—Number of seconds that the router waits between the transmission of loopback cells when it is verifying the state of a PVC
- OAM alarm down count—Number of successive alarm cells that the router receives before reporting that a PVC is down
- OAM alarm clear time out—Number of seconds that the router waits before reporting that a PVC is up after the PVC has stopped receiving alarm cells
- OAM continuity check—Status of F5 OAM continuity check verification on the PVC: enabled or disabled
- Inverse ARP—Status of Inverse ARP (InARP) on the PVC: enabled or disabled

- Example 1

```
host1#show atm vc-class
premium-subscriber-class
dsl-subscriber-class
found 2 VC class entrie(s) in the system
```

- Example 2

```
host1#show atm vc-class premium-subscriber-class
Encapsulation Type                  :AUTO
Service Category                    :CBR
Peak Cell Rate                      :200 kbps
OAM VC Integrity                    :enabled
OAM VC Integrity loop-back timer    :60 seconds
OAM alarm down count                :5
```

- Example 3

```
host1#show atm vc-class premium-subscriber-class include-defaults
Encapsulation Type                  :AUTO
Service Category                    :CBR
Peak Cell Rate                      :200 kbps
OAM VC Integrity                    :enabled
OAM VC Integrity loop-back timer    :60 seconds
OAM VC Integrity Up Retry Count     :3
OAM VC Integrity Down Retry Count   :5
OAM VC Integrity Retry Frequency    :1
OAM alarm down count                :5
OAM alarm clear time out            :3 seconds
OAM continuity check                :disabled
Inverse ARP                         :disabled
```

show atm vp

- Use to display detailed statistics for a specific ATM VP configured on the router.
- Field descriptions
  - ServiceCategory—Service type on the VP tunnel, if configured: UBR, UBR-PCR, VBR-NRT, VBR-RT, or CBR
  - Peak Cell Rate—Peak cell rate in kilobits per second, if a VP tunnel is configured
  - Maximum VCI per VPI—Maximum number of virtual circuits on each virtual path
  - Current VCs—Number of VCs currently configured on the router
  - InPackets—Number of packets received
  - InBytes—Number of bytes received
  - InCells—Number of ATM cells received
  - OutPackets—Number of packets transmitted
  - OutBytes—Number of bytes transmitted
  - OutCells—Number of ATM cells transmitted
  - InErrors—Number of packets with errors received
  - OutErrors—Number of packets not transmitted on this VP due to errors
  - InPacketDiscards—Number of incoming packets discarded
  - InPacketUnknownProtocol—Number of incoming packets with an unknown protocol type
  - InByteDiscards—Number of incoming bytes discarded
  - CrcErrors—Number of CRC errors detected
  - SAR time-outs—Number of segmentation and reassembly (SAR) timeouts reached
  - Over-sized SDUs—Number of oversized service data units (SDUs) received
  - The following fields appear only if F4 OAM is enabled on the VP:
    - Sending End to End Loopback Cells—Enabled, Disabled
    - End to End OAM CC verification—Enabled, Disabled
    - VP State—State of the VP: up, down
    - VP Oam State—OAM state of the VP: not managed (normal OAM state with no OAM fault conditions), AIS, RDI
    - InOamF4EndCells—Number of F4 end-to-end cells received
    - InOamF4EndCellsDropped—Number of incoming F4 end-to-end cells that were dropped
    - InOamF4EndLoopbackCells—Number of F4 end-to-end loopback cells received
    - InOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands received

- InOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses received
- InOamF4EndAisCells—Number of F4 end-to-end AIS cells received
- InOamF4EndRdiCells—Number of F4 end-to-end RDI cells received
- InOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells received
- InOamF4EndCCCells—Number of F4 end-to-end CC cells received
- OutOamF4EndCells—Number of F4 end-to-end CC cells transmitted
- OutOamF4EndLoopbackCells—Number of F4 end-to-end loopback cells transmitted
- OutOamF4EndLoopbackCommands—Number of F4 end-to-end loopback commands transmitted
- OutOamF4EndLoopbackResponses—Number of F4 end-to-end loopback responses transmitted
- OutOamF4EndRdiCells—Number of F4 end-to-end RDI cells transmitted
- OutOamF4EndCCActDeActCells—Number of F4 end-to-end activation or deactivation CC cells transmitted
- OutOamF4EndCCCells—Number of F4 end-to-end CC cells transmitted
- Time since last status change—Time since last reported change to the end-to-end OAM circuit status
- Segment OAM CC verification—Enabled or Disabled
- VP State—State of the VP: up, down
- VP Oam State—OAM state of the VP: not managed (normal OAM state with no OAM fault conditions), AIS, RDI
- InOamF4SegmentCells—Number of F4 segment cells received
- InOamF4SegmentCellsDropped—Number of incoming F4 segment cells that were dropped
- InOamF4SegmentLoopbackCells—Number of F4 segment loopback cells received
- InOamF4SegmentLoopbackCommands—Number of F4 segment loopback commands received
- InOamF4SegmentLoopbackResponses—Number of F4 segment loopback responses received
- InOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells received
- InOamF4SegCCCells—Number of F4 segment CC cells received
- OutOamF4SegmentCells—Number of F4 segment cells transmitted
- OutOamF4SegmentLoopbackCells—Number of F4 segment loopback cells transmitted
- OutOamF4SegmentLoopbackCommands—Number of F4 segment loopback commands transmitted

- OutOamF4SegmentLoopbackResponses—Number of F4 segment loopback responses transmitted
- OutOamF4SegRdiCells—Number of F4 segment RDI cells transmitted
- OutOamF4SegCCActDeActCells—Number of F4 segment activation or deactivation CC cells transmitted
- OutOamF4SegCCCells—Number of F4 segment CC cells transmitted
- Time since last status change—Time since last reported change to the segment OAM circuit status

VP Description—Text description for this VP, if configured

- Example

```
host1#show atm vp atm 12/0 1
Maximum VCI per VPI: 65535     Current VCs: 3
InPackets                        :1604710953
InBytes                          :205403001984
InCells                          :519165564
OutPackets                       :1604632002
OutBytes                         :205392896256
OutCells                         :4813896006
InErrors                         :0
OutErrors                        :0
InPacketDiscards                 :0
InPacketUnknownProtocol          :0
InByteDiscards                   :0
CrcErrors                        :0
SAR time-outs                    :0
Over-sized SDUs                  :0
Sending End To End Loopback Cells Disabled:
End To End OAM CC verification Disabled
VP State                         :up
VP Oam State                     :not managed
InOamF4EndCells                  :0
InOamF4EndCellsDropped           :0
InOamF4EndLoopbackCells          :0
  InOamF4EndLoopbackCommands       :0
  InOamF4EndLoopbackResponses      :0
InOamF4EndAisCells               :0
InOamF4EndRdiCells               :0
InOamF4EndCCActDeActCells        :0
InOamF4EndCCCells                :0
OutOamF4EndCells                 :0
OutOamF4EndLoopbackCells         :0
  OutOamF4EndLoopbackCommands      :0
  OutOamF4EndLoopbackResponses     :0
OutOamF4EndRdiCells              :0
OutOamF4EndCCActDeActCells       :0
OutOamF4EndCCCells               :0
Time since last status change    :08:48:43
Segment OAM CC verification Disabled
VP State                         :up
VP Oam State                      :not managed
InOamF4SegmentCells              :0
InOamF4SegmentCellsDropped       :0
InOamF4SegmentLoopbackCells      :0
  InOamF4SegmentLoopbackCommands   :0
  InOamF4SegmentLoopbackResponses  :0
InOamF4SegCCActDeActCells        :0
InOamF4SegCCCells                :0
```

```
OutOamF4SegmentCells             :0
OutOamF4SegmentLoopbackCells     :0
  OutOamF4SegmentLoopbackCommands  :0
  OutOamF4SegmentLoopbackResponses :0
OutOamF4SegRdiCells              :0
OutOamF4SegCCActDeActCells       :0
OutOamF4SegCCCells               :0
Time since last status change    :08:48:44
VP Description: ATM-12/0-VPI-1
```

show atm vp-description

- Use to display VP descriptions configured using the **atm vp-description** command.
- To display all VP descriptions configured on the router, issue the command without an ATM identifier or VPI number (Example 1).
- To display all VP descriptions for a particular ATM interface, specify the ATM interface identifier without the VPI number (Example 2).
- To display the VP description for a particular VPI, specify both the ATM interface identifier and the VPI number (Example 3).
- Field descriptions
  - Interface—ATM interface identifier
  - VPI—Virtual path identifier
  - Description—Text description configured for the VP
- Example 1—Displays all VP descriptions configured on the router

```
host1#show atm vp-description
Interface    VPI    Description
ATM 2/0      0      atm20Vpi0Subscribers
ATM 2/0      1      atm20Vpi1Subscribers
ATM 2/1      0      atm21Vpi0Subscribers
```

- Example 2—Displays all VP descriptions for the specified ATM interface

```
host1#show atm vp-description atm 2/0
Interface    VPI    Description
ATM 2/0      0      atm20Vpi0Subscribers
ATM 2/0      1      atm20Vpi1Subscribers
```

- Example 3—Displays the VP description for the specified VPI

```
host1#show atm vp-description atm 2/0 1
Interface    VPI    Description
ATM 2/0      1      atm20Vpi1Subscribers
```

show atm vp-tunnel

- Use to display a summary of all configured ATM virtual path tunnels.
- Field descriptions
  - Intfc—Interface number
  - VPI—Virtual path identifier
  - Type—VP tunnel traffic management type
  - Kbps—Rate, in Kbps
  - Description—Text description for the VP, if configured
- Example

```
host1#show atm vp-tunnel 9/1
Intfc    VPI  Type  Kbps    Description
ATM 9/1  2    Cbr   4096    atm91Vpi2Subscribers
```

show mpls cross-connects atm

- Use to display all ATM cross-connects (passthrough connections between local subinterfaces).
- See *Monitoring ATM Cross-Connects for Layer 2 Services over MPLS* in *JUNOSe BGP and MPLS Configuration Guide, Chapter 6, Monitoring Layer 2 Services over MPLS* for information about using the **show mpls cross-connects atm** command.

show nbma arp

- Use to display ARP table entries for ATM NBMA interfaces.
- Field descriptions
  - IP Address—IP address of the entry
  - VPI/VCI—VPI and VCI of the entry
  - Interface—Interface specifier of the entry
- Example

```
host1#show nbma arp
                NBMA ARP Table Entries
IP Address              VPI/VCI         Interface
1.1.1.2                 0/100           4/1
2.2.2.2                 0/101           4/1
```

Chapter 2

Configuring Frame Relay

This chapter describes how to configure a Frame Relay interface on E-series routers.

This chapter contains the following sections:

Overview

Frame Relay is a public, connection-oriented packet service based on the core aspects of the Integrated Services Digital Network (ISDN). Frame Relay eliminates all processing at the network layer and greatly restricts data-link layer processing. Such simplified processing is possible because of the availability of virtually error-free physical connections and the presence of intelligent protocols at the end-user site, which can detect and retransmit faulty or discarded packets.

Frame Relay shifts responsibility for error recovery and flow control to the end user, thereby reducing protocol complexity and allowing high-speed packet delivery with low transit delay.

For a list of the modules on which you can configure Frame Relay, see *ERX Module Guide, Appendix A, Module Protocol Support*.

Framing

E-series routers support the following framing features:

- HDLC for data-link framing
- 2-byte addresses only
- 8188-byte information field size (8192 minus 2 bytes for the address and a 16-bit CRC) or 8186-byte information field size (8192 minus 2 bytes for the address and a 32-bit CRC)

The router does not support:

- Protocol-dependent fragmentation
- Autodetection of the Local Management Interface (LMI) protocol type

Error Frames

The router relies on higher-layer protocols to detect and recover from Frame Relay data loss. All Frame Relay error frames are discarded.

Unicast and Multicast Addressing

Most Frame Relay services support both unicast (individual) and multicast (group) addressing. Under the most common implementation of multicasting, the Frame Relay network maps multiple individual addresses to a single multicast data-link connection identifier (DLCI) and delivers copies of a single Frame Relay packet to each member of the group.

NOTE: The E-series router supports only unicast addressing.

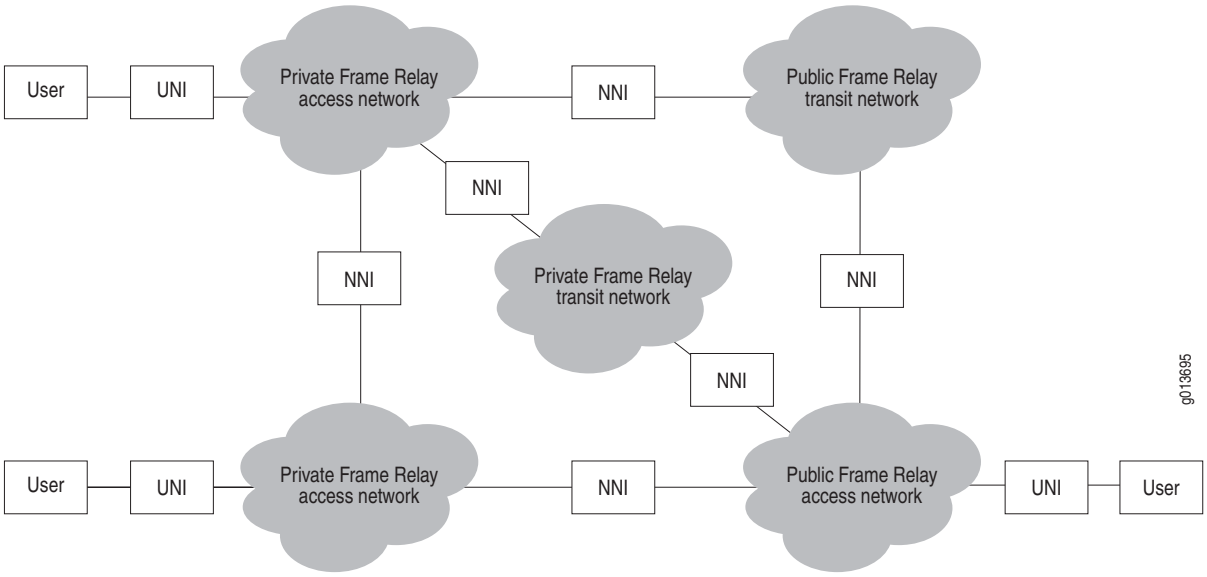
User-to-Network and Network-to-Network Interfaces

The Frame Relay User-to-Network Interface (UNI) is a protocol that permits users to access private or public Frame Relay networks and to establish a communications path to another user within the same network.

The Network-to-Network Interface (NNI) makes connections possible between users connected to different Frame Relay networks. These separate Frame Relay networks can be considered as subnetworks within a complete network service.

Figure 4 shows the interconnection of these types of subnetworks and the location of NNI between them.

Figure 4: Interconnection and Relationship of NNIs and Subnetworks



Platform Considerations

You can configure Frame Relay interfaces on the following E-series routers:

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

NOTE: The E120 router and the E320 router do not support configuration of Frame Relay interfaces.

Module Requirements

For information about the modules that support Frame Relay interfaces on ERX-14xx models, ERX-7xx 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 Frame Relay.

Interface Specifiers

The interface specifier format that you use depends on the type of physical interface on which you want to configure Frame Relay.

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about Frame Relay, consult the following resources:

- RFC 2115—Management Information Base for Frame Relay DTEs Using SMIv2 (September 1997)
- RFC 2863—The Interfaces Group MIB (June 2000)
- RFC 2427—Multiprotocol Interconnect over Frame Relay (September 1998)
- Frame Relay Forum—User-to-Network Implementation Agreement (UNI), FRF 1.1 (January 1996)
- Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997)
- ANSI T1.617 Annex D
- ITU-T Recommendation Q.922, Integrated Services Digital Network (ISDN) Data Link Layer Specification for Frame Mode Bearer Services; Annex A (February 1992)
- ITU-T Q.933 Annex A

Before You Configure Frame Relay

Before you attempt to configure a Frame Relay interface, configure the physical line interface over which Frame Relay traffic flows.

This process is described in the following chapters:

- *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*

The procedures described in this chapter assume that a physical interface has been configured.

Configuring Frame Relay

Configure a Frame Relay interface by entering Interface Configuration mode. The procedure that follows is an example of a Frame Relay configuration on a serial interface. All tasks are mandatory unless otherwise noted.

To configure a Frame Relay interface:

1. From Configuration mode, enter the physical interface on which you want to configure Frame Relay.

   host1(config)#**interface serial 3/1:2/1**

2. Select Frame Relay as the encapsulation method for the interface.

   host1(config-if)#**encapsulation frame-relay ietf**

3. (Optional) Assign a text description or an alias to the major interface.

   host1(config-if)#**frame-relay description boston01**

4. (Optional) Enable SNMP link status processing on the major interface.

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

5. Configure the interface as a DTE, DCE, or NNI.

   host1(config-if)#**frame-relay intf-type dte**

6. Configure the LMI type.

   host1(config-if)#**frame-relay lmi-type ansi**

7. (Optional) Configure Frame Relay counters and timers.

   host1(config-if)#**frame-relay lmi-n391dte 20**

8. Configure the cyclic redundancy check (CRC).

   host1(config-if)#**crc 32**

9. Create a subinterface.

   host1(config)#**interface serial 3/1:2/1.1**

10. (Optional) Assign a text description or an alias to the subinterface.

    host1(config-subif)#**frame-relay description westford011**

11. (Optional) Enable SNMP link status processing on the subinterface.

    host1(config-subif)#**snmp trap frame-relay link-status**

12. Add a circuit to a subinterface.

    host1(config-subif)#**frame-relay interface-dlci 17 ietf**

13. Assign a local IP address to the circuit.

    host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

14. (Optional) Use **show** commands to verify that your configuration changes are correct by checking the state of the interfaces.

    host1#**show frame-relay lmi**
    host1#**show frame-relay map**
    host1#**show frame-relay pvc**

15. (Optional) Disable the local management interface.

    host1#**no frame-relay keepalive**

16. (Optional) Disable the interface.

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

crc

- Use to set the number of bits used for CRC.
- The CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data.
- 16 and 32 indicate the number of bits per frame that are used to calculate the frame check sequence (FCS).
- A 32-bit CRC transmits longer streams at faster rates and therefore provides better ongoing error detection, such as for an OC12/STM4 POS module.
- The default is 16. You must configure CRC (CRC16 or CRC32) to match the configuration on the other side of the Frame Relay connection.
- Example

  host1(config-if)#**crc 32**

- Use the **no** version to set the CRC to the default value.

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**

- Use the **no** version to remove Frame Relay configuration from an interface.

frame-relay description

- Use to assign a text description or an alias to a Frame Relay interface or subinterface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use the **show frame-relay interface** or **show frame-relay subinterface** command to display the text description.
- Examples

  host1(config-if)#**frame-relay description boston01**
  host1(config-subif)#**frame-relay description toronto011**

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

frame-relay interface-dlci ietf

- Use to configure a Frame Relay permanent virtual circuit (PVC) over a subinterface.
- The **ietf** keyword is mandatory and indicates RFC 2427 encapsulation.
- Define a DLCI in the range 16–1007.
- To configure a Frame Relay PVC, you must specify a DLCI.
- Frame Relay service is offered in the form of PVCs. A PVC is a data-link connection that is predefined on both ends of the connection. A network operator assigns the endpoints of the circuit. Although the actual path taken through the network may vary from time to time, the beginning and end of the circuit do not change. This type of circuit behaves like a dedicated point-to-point circuit.
- PVCs are identified by DLCIs. A DLCI is a 10-bit channel number that is attached to data frames to tell a Frame Relay network how to route the data. Frame Relay is *statistically multiplexed*, which means that only one frame can be transmitted at a time, but many logical connections can coexist on a single physical line. The DLCI allows the data to be logically tied to one of the connections, so that when the data gets to the network, the network knows where to send it.
- DLCIs on the same physical line must match. However, DLCIs have local significance; that is, if the DLCIs are not on the same physical line, the end devices at two different ends of a connection may use a different DLCI to refer to the same connection.
- The router does not support switched virtual circuits (SVCs). An SVC is an any-to-any connection that can be established or removed as needed. With SVCs, you initiate calls using Frame Relay by requesting a destination address and assigning a DLCI, which is established for the duration of the call.
- Example

  host1(config-subif)#**frame-relay interface-dlci 17 ietf**

- Use the **no** version to remove DLCI/PVC assignment.

frame-relay intf-type

- Use to configure a Frame Relay interface circuit to operate as data communications equipment (DCE), data terminal equipment (DTE), or NNI.
- Frame Relay provides packet-switching data communications between user devices and network equipment across the interface. User devices are referred to as DTE.
- Network equipment that interfaces with a DTE is referred to as a DCE.
- NNI provides a connection between two Frame Relay subnetworks.
- If your router is connected to a Frame Relay switch, configure the interface as a DTE. If your router is connected by a point-to-point line, configure one end as the DTE and the other as the DCE.
- Example

  host1(config-if)#**frame-relay intf-type dte**

- Use the **no** version to set the default of DTE.

frame-relay keepalive

- Use to enable the LMI on the interface.
- You can specify the keepalive interval in seconds.
- Make sure the value on the DTE is less than the value set on the DCE.
- The default is 10 seconds.
- Example

  host1#**no frame-relay keepalive**

- Use the **no** version to disable LMI on the interface.

frame-relay lmi-n391dte
frame-relay lmi-n392dce
frame-relay lmi-n392dte
frame-relay lmi-n393dce
frame-relay lmi-n393dte
frame-relay lmi-t391dte
frame-relay lmi-t392dce

- Use to configure LMI counters and timers.
- LMI counters and timers have configurable ranges that allow you to control the state of the Frame Relay interface. In general, accept the default values for the timers and counters, unless you need to modify them according to a special arrangement with your customers.
- Some commands have DTE and DCE versions. Use the **dte** version of the command if the interface is operating as a DTE. Use the **dce** version of the command if the interface is operating as a DCE. Use both versions of the command if the interface is operating as an NNI.

- Use the **frame-relay lmi-n391dte** command to set the N391 full-status polling counter. When you set this counter to a number, *n*, the *n*th request is a full-status request. The range is 1–255 event messages. The default is 6 event messages.
- Use the **frame-relay lmi-n392dte** or **frame-relay lmi-n392dce** command to set the N392 error threshold counter, which specifies the number of errors within N393 events that will place the interface in an operationally down state. The range is 1–10. The default for the DTE version is 3. The default for the DCE version is 2.
- Use the **frame-relay lmi-n393dte** or **frame-relay lmi-n393dce** command to set the N393 monitored events counter to specify the diagnostic window used to verify link integrity. Detection of N392 errors within the window of N393 samples places the interface in an operationally down state. The range is 1–10 events. The default for the DTE version of the command is 4 events. The default for the DCE version is 2 events.
- Use the **frame-relay lmi-t391dte** command to set the T391 link integrity polling timer interval between status inquiries issued by the DTE. The network checks that the DTE polls within the verification interval, T392. The range is 5–30 seconds. The default is 10 seconds.
- Use the **frame-relay lmi-t392dce** command to set the T392 polling verification timer that specifies the maximum interval (in seconds) between the receipt of status inquiries from the DTE equipment before it considers it as an error event. The range is 5–30 seconds. The default is 15 seconds.
- Example

  host1(config-if)#**frame-relay lmi-n391dte 20**

- Use the **no** version to remove the current setting and set the default.

frame-relay lmi-type

- Use to configure one of the local management interface types.
- LMI provides configuration and status information relating to the virtual circuits operating over Frame Relay.
- LMI specifies polling mechanisms to receive incremental and full-status updates from the network.
- E-series routers conform to the following LMI specifications:
  - **ansi**—ANSI T1.617 Annex D
  - **q933a**—ITU-T Q.933 Annex A
  - **cisco**—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
  - **none**—Suppresses LMI
- The default is **cisco**.
- Example

  host1(config-if)#**frame-relay lmi-type ansi**

- Use the **no** version to return to the default LMI type.

interface pos

- Use to configure a POS interface in *slot/port* format:
  - *slot*—Router chassis slot
  - *port*—Line module port
- Example

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

- Use the **no** version to remove the POS interface.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Frame Relay. For example, for a channelized T3 interface use *slot/port:channel/subchannel*.
- Use to configure a Frame Relay subinterface in the appropriate format by selecting a previously configured physical interface. For example, for a T3-Frame interface use *slot/port.subinterface;* for a channelized T1/channelized E1 interface use *slot/port.channel-group.subinterface*.

NOTE: Before you configure Frame Relay, see the appropriate chapter in this guide for details on configuring physical interfaces.

  - *slot*—Router chassis slot
  - *port*—CT3, T3, or E3 module I/O port
  - *channel*—T1 (DS1) channel
  - *subchannel*—Set of DS0 timeslots. See the *Fractional T1* section in *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
  - *subinterface*—User-assigned nonnegative number that identifies a Frame Relay subinterface
- Example

  host1(config-if)#**interface serial 3/1:2/1.1**

- Use the **no** version to remove the subinterface or the serial interface.

ip address

- Use to assign an IP address and subnet mask to a subinterface.
- Example

  host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

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

pos description

- Use to assign a text description or an alias to a POS HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use the **show interfaces pos** command to display the text description. For details, see *Monitoring POS* in *Chapter 9, Configuring Packet over SONET*.
- Example

  host1(config-if)#**pos description austin01 pos interface**

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

serial description

- Use to assign a text description or an alias to a serial HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use the **show interfaces serial** command to display the text description. For example, for a channelized T3 interface, see *Monitoring Interfaces* in *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*.
- Example

  host1(config-if)#**serial description ottawa012 hdlc channel**

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

shutdown

- Use to disable a Frame Relay interface.
- Example

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

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

snmp trap frame-relay link-status

- Use to enable SNMP link status processing for a Frame Relay major interface or subinterface.
- To enable SNMP link status processing for a Frame Relay major interface, you must issue the command from Interface Configuration mode.
- To enable SNMP link status processing for a Frame Relay subinterface, you must issue the command from Subinterface Configuration mode.

- Examples

  host1(config-if)#**snmp trap frame-relay link-status**
  host1(config-subif)#**snmp trap frame-relay link-status**

- Use the **no** version to disable SNMP link status processing for a Frame Relay major interface or subinterface.

End-to-End Fragmentation and Reassembly

The fragmentation and reassembly feature reduces excessive delays of Frame Relay packets by breaking them up into smaller fragments and interleaving them with real-time frames. By doing this, real-time and non-real-time data frames can be carried together on lower-speed links without causing excessive delays to the real-time traffic. On receiving the smaller fragments by the peer interface, the fragments are reassembled into their original packet. For example, short delay-sensitive packets, such as packetized voice, can race ahead of larger delay-insensitive packets, such as common data packets.

E-series routers support end-to-end fragmentation according to the FRF.12 Implementation Agreement standard. Unlike UNI and NNI fragmentation, end-to-end supports fragmentation only at the endpoints. End-to-end fragmentation and reassembly are supported only on non-multilink Frame Relay interfaces on cOC12/STM4 and CT3 12 FO modules.

You configure end-to-end fragmentation at the Frame Relay subinterface level. Fragmentation is applied to all PVCs associated with the subinterface. In most cases, fragmentation and reassembly are used together. Fragmentation and reassembly, however, can be configured separately for each map class.

For additional information, see Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997).

Frame Fragmentation

When you enable fragmentation, you can specify a maximum payload size of the resulting fragments. If the maximum payload size is not specified, the default value of 52 bytes is used. When enabled, fragmentation begins when the portion of the packet that has not been transmitted in previous fragments exceeds the configured maximum payload size. The fragmentation process continues until the entire packet has been transmitted. Frames that do not exceed the configured maximum payload size are not fragmented.

If you disable fragmentation, all packets transmitted by the Frame Relay subinterface are transmitted intact.

Frame Reassembly

When reassembly is disabled and a data frame is received, a few scenarios may occur:

- If the frame is not a fragment, it is forwarded normally.
- If the frame is a fragment and the upper interface is IP (that is, the interface above the Frame Relay subinterface), then the fragment is immediately discarded.

If you enable reassembly, then received fragments undergo the reassembly process. Packets that are not fragments are forwarded as normal.

Map Class

Within Frame Relay, a map class acts as a container or context for fragmentation and reassembly parameters. Within the map class context, you can explicitly enable fragmentation and reassembly.

After you define a map class, you can apply it to an unlimited number of subinterfaces. This allows you to change fragmentation and reassembly parameters one time and have the changes immediately reflected in all subinterfaces configured to use that map class.

Configuring End-to-End Fragmentation

You configure end-to-end fragmentation and reassembly on a subinterface in much the same way you configure a standard Frame Relay interface. In this example, end-to-end fragmentation and reassembly is configured on a single subinterface with a 100-byte fragment size (maximum payload size). All tasks are mandatory unless otherwise noted.

NOTE: The procedure described in this section assumes that a physical interface has been configured. See *Before You Configure Frame Relay* on page 104.

To configure end-to-end fragmentation and reassembly:

1. Create a map class that you can apply to subinterfaces.

   host1(config)#**map-class frame-relay testmap**

2. Specify fragmentation and reassembly for the map class. Optionally, you can specify the maximum payload size of a fragment.

   host1(config-map-class)#**frame-relay fragment 100**

3. Enter the physical interface on which you want to configure Frame Relay end-to-end fragmentation and reassembly.

   host1(config-map-class)#**interface serial 5/0:4/1**

4. Select Frame Relay as the encapsulation method for the interface.

   host1(config-if)#**encapsulation frame-relay ietf**

5. Create a subinterface.

   host1(config-if)#**interface serial 5/0:4/1.1**

6. Add a circuit to a subinterface.

   host1(config-subif)#**frame-relay interface-dlci 16 ietf**

7. Assign a local IP address to the circuit.

   host1((config-subif)#**ip address 42.42.42.41 255.255.255.0**

8. Associate a map class with a subinterface.

   host1(config-subif)#**frame-relay class testmap**

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**

- Use the **no** version to remove Frame Relay configuration from an interface.

frame-relay class

- Use to associate a map class with a subinterface.
- Example

  host1(config-subif)#**frame-relay class testmap**

- Use the **no** version to remove the association between the subinterface and the specified map class from the subinterface.

frame-relay fragment

- Use to configure fragmentation and reassembly for the map class.
- Specify the keyword **fragmentation-only** to specify only fragmentation, so that reassembly is not performed.
- Specify the keyword **reassembly-only** to specify only reassembly, so that fragmentation is not performed.
- Specify the maximum payload size of a fragment by using a value from 16–8188 bytes. If a value is not specified, the default value of 52 bytes is used.
- Make sure the value for the maximum payload size of a fragment is less than or equal to the MTU size, otherwise fragmentation never occurs.
- Make sure the maximum payload size is larger than any voice packet so that voice frames are not fragmented.

- Examples

  host1(config-map-class)#**frame-relay fragment 100**
  host1(config-map-class)#**frame-relay fragment fragmentation-only**

- Use the **no** version to stop fragmentation and reassembly on the subinterface.

frame-relay interface-dlci ietf

- Use to configure a Frame Relay PVC over a subinterface.
- The **ietf** keyword is mandatory and indicates RFC 2427 encapsulation.
- Define a DLCI in the range 16–1007.
- To configure a Frame Relay PVC, you must specify a DLCI.
- Frame Relay service is offered in the form of PVCs. A PVC is a data-link connection that is predefined on both ends of the connection. A network operator assigns the endpoints of the circuit. Although the actual path taken through the network may vary from time to time, the beginning and end of the circuit do not change. This type of circuit behaves like a dedicated point-to-point circuit.
- PVCs are identified by DLCIs. A DLCI is a 10-bit channel number that is attached to data frames to tell a Frame Relay network how to route the data. Frame Relay is *statistically multiplexed*, which means that only one frame can be transmitted at a time, but many logical connections can coexist on a single physical line. The DLCI allows the data to be logically tied to one of the connections, so that when the data gets to the network, the network knows where to send it.
- DLCIs on the same physical line must match. However, DLCIs have local significance; that is, if the DLCIs are not on the same physical line, the end devices at two different ends of a connection may use a different DLCI to refer to the same connection.
- The router does not support SVCs. An SVC is an any-to-any connection that can be established or removed as needed. With SVCs, you initiate calls using Frame Relay by requesting a destination address and assigning a DLCI, which is established for the duration of the call.
- Example

  host1(config-subif)#**frame-relay interface-dlci 16 ietf**

- Use the **no** version to remove DLCI/PVC assignment.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Frame Relay. For example, for a channelized T3 interface use *slot/port:channel/subchannel*.
- Use to configure a Frame Relay subinterface in the appropriate format by selecting a previously configured physical interface. For example, for a T3-Frame interface use *slot/port.subinterface;* for a channelized T1/channelized E1 interface use *slot/port.channel-group.subinterface*.

NOTE: See *Before You Configure Frame Relay* on page 104 for more information about configuring the underlying physical interfaces.

  - *slot*—Router chassis slot
  - *port*—CT3, T3, or E3 module I/O port
  - *channel*—T1 (DS1) channel
  - *subchannel*—Set of DS0 timeslots; for information, see *Fractional T1* in *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
  - *subinterface*—User-assigned nonnegative number that identifies a Frame Relay subinterface
- Example

  host1(config-if)#**interface serial 5/0:4/1.1**

- Use the **no** version to remove the subinterface or the serial interface.

ip address

- Use to assign an IP address and subnet mask to a subinterface.
- Example

  host1((config-subif)#**ip address 42.42.42.41 255.255.255.0**

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

map-class frame-relay

- Use to create a map class.
- Example

  host1(config)#**map-class frame-relay testmap**

- Use the **no** version to remove a map class.

Monitoring Frame Relay

Use the **show frame-relay** commands described in this section to monitor Frame Relay interfaces.

You can set a statistics baseline for Frame Relay interfaces, subinterfaces, or circuits using the **baseline frame-relay** command.

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 *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface* for details.

If you do not specify an interface type for the appropriate **show** command, the output indicates whether a serial or POS interface is being displayed.

baseline frame-relay interface

- Use to set a statistics baseline at the Frame Relay layer for multilink Frame Relay, POS, serial or GRE tunnel interfaces, subinterfaces, or circuits.
- 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.
- Specify an interface or subinterface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide.*
- Specify a circuit using the interface type and specifier and the **dlci** keyword and the *dlci* number.
- You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set baselines individually.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message "No baseline has been set" is displayed instead.
- The regular interface statistics and LMI statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines.
- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

```
host1#baseline frame-relay interface serial 2/0:1/1

host1#show frame-relay interface delta
Frame relay interface 2/0:1/1, status is lowerLayerDown
Number of interface down transitions is 0
Time since last status change 21:06:34
Number of configured circuits: 0
Time since last baseline 00:00:05
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
```

- There is no **no** version.

show frame-relay interface

- Use to display statistics for the Frame Relay layer in a multilink Frame Relay, POS, serial, or GRE tunnel interface.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
- Use the **brief** keyword to display the operational status of all configured interfaces.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - status—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Description—Text description or alias if configured for the interface
  - In bytes—Number of inbound bytes received on the interface
  - Out bytes—Number of outbound bytes transmitted on the interface
  - In frames—Number of inbound frames received on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - In errors—Number of inbound errors received on the interface
  - Out errors—Number of outbound errors transmitted on the interface
  - In discards—Number of inbound packets discarded
  - Out discards—Number of outbound packets discarded
  - In unknown protos—Number of packets received on the interface with unknown protocols
  - Time since last status change—Time since the last status change on the interface

- Example

```
host1#show frame-relay interface
Frame relay interface 3/2:1/1, status is up
Description: boston01
Time since last status change 01:21:10
  In bytes: 19712            Out bytes: 60918
  In frames: 1232            Out frames: 1232
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
Frame relay interface 3/2:2/1, status is up
Description: newyork02
Time since last status change 03:06:18
  In bytes: 19728            Out bytes: 60702
  In frames: 1233            Out frames: 1233
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
Frame relay interface 3/2:3/1, status is up
Description: chicago03
Time since last status change 01:20:38
  In bytes: 19696            Out bytes: 60744
  In frames: 1231            Out frames: 1231
  In errors: 0               Out errors: 0
```

show frame-relay lmi

- Use to display configuration and state information and statistics about the LMI for a multilink Frame Relay, POS, serial, or GRE tunnel interface.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - This command displays both DTE and DCE fields for NNI.
  - For the DTE:
    - Enquiries sent—Total number of LMI status enquiries sent by the DTE on this interface
    - Full enquiries sent—Total number of LMI full-status enquiries sent by the DTE on this interface
    - Enquiry responses received—Total number of LMI full- and regular-status responses received by the DTE on this interface
    - Full enquiry responses received—Total number of LMI full-status responses received by the DTE on this interface
    - Async updates received—Total number of LMI asynchronous updates received by the DTE on this interface
    - Unknown messages received—Total number of unknown LMI messages received on this interface

- Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DTE on this interface
- No response timeouts—Total number of times a timeout occurred without receiving a response to an LMI request by the DTE on this interface
- Last sequence number sent—Last sequence number sent on this interface
- Last sequence number received—Last sequence number received on this interface

- For the DCE:
  - Enquiries received—Total number of LMI status enquiries received by the DCE on this interface
  - Enquiry responses sent—Total number of LMI status responses sent by the DCE on this interface
  - Full enquiry responses sent—Total number of LMI full-status responses sent by the DCE on this interface
  - Async updates sent—Total number of LMI asynchronous updates sent by the DCE on this interface
  - Unknown messages received—Total number of unknown LMI messages received on this interface
  - Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DCE on this interface
  - No response timeouts—Total number of times a timeout occurred without receiving a status inquiry from the DTE on this interface
  - Last sequence number sent—Last sequence number sent on this interface
  - Last sequence number received—Last sequence number received on this interface

- Example

```
host1#show frame-relay lmi
LMI information for frame relay NNI interface 3/2:1/1
DTE Parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
DCE Parameter N392 is 2, N393 is 2, T392 is 15
Configured LMI type is ANSI, status is up
Time since last status change 01:21:14
  Enquiries received: 1232
  Full enquiries received: 207
  Enquiry responses sent: 1232
  Full enquiry responses sent: 207
  Async updates sent: 0
  Unknown messages received: 0
  Loss of sequencing detected: 2
  No response timeouts: 0
  Last sequence number sent: 0
  Last sequence number received: 0
  Unknown messages received: 0
  Loss of sequencing detected: 2
```

```
LMI information for frame relay DCE interface 3/2:2/1
Parameter N392 is 2, N393 is 2, T392 is 15
Configured LMI type is ANSI, status is up
Time since last status change 03:06:22
  Enquiries received: 1233
  Full enquiries received: 207
  Enquiry responses sent: 1233
  Full enquiry responses sent: 207
  Async updates sent: 0
  Last sequence number sent: 0
  Last sequence number received: 0
```

show frame-relay map

- Use to display the current Frame Relay map entries and information about Frame Relay connections.
- Field descriptions
  - Frame relay sub-interface—Interface number and one of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - DLCI—Provides decimal value, hexadecimal value, and its value as it appears on the wire
- Example

```
host1#show frame-relay map
Frame relay sub-interface 3/2:1/1.1 (up): DLCI 101(0x65,0x58)
Frame relay sub-interface 3/2:1/1.2 (up): DLCI 102(0x66,0x78)
Frame relay sub-interface 3/2:1/1.3 (up): DLCI 103(0x67,0x78)
Frame relay sub-interface 3/2:1/1.4 (up): DLCI 104(0x68,0x98)
Frame relay sub-interface 3/2:1/1.5 (up): DLCI 105(0x69,0x98)
Frame relay sub-interface 3/2:1/1.6 (up): DLCI 106(0x6a,0xb8)
Frame relay sub-interface 3/2:1/1.7 (up): DLCI 107(0x6b,0xb8)
Frame relay sub-interface 3/2:1/1.8 (up): DLCI 108(0x6c,0xd8)
Frame relay sub-interface 3/2:1/1.9 (up): DLCI 109(0x6d,0xd8)
Frame relay sub-interface 3/2:1/1.10 (up): DLCI 110(0x6e,0xf8)
Frame relay sub-interface 3/2:1/1.11 (up): DLCI 111(0x6f,0xf8)
Frame relay sub-interface 3/2:1/1.12 (up): DLCI 112(0x70,0x1c)
Frame relay sub-interface 3/2:1/1.17 (up): DLCI 117(0x75,0x5c)
Frame relay sub-interface 3/2:1/1.18 (up): DLCI 118(0x76,0x7c)
Frame relay sub-interface 3/2:1/1.19 (up): DLCI 119(0x77,0x7c)
Frame relay sub-interface 3/2:1/1.20 (up): DLCI 120(0x78,0x9c)
Frame relay sub-interface 3/2:1/1.21 (up): DLCI 121(0x79,0x9c)
Frame relay sub-interface 3/2:1/1.22 (up): DLCI 122(0x7a,0xbc)
Frame relay sub-interface 3/2:1/1.23 (up): DLCI 123(0x7b,0xbc)
Frame relay sub-interface 3/2:1/1.24 (up): DLCI 124(0x7c,0xdc)
```

show frame-relay pvc

- Use to display statistics about PVCs for Frame Relay layer on a multilink Frame Relay, POS, serial, or GRE tunnel interface or a specific PVC.
- Optionally, you can specify an interface using the interface type and specifier. For more information, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide.*
- Specify a virtual circuit using the DLCI number.
- Use the **brief** keyword to display abbreviated PVC information.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - DLCI—DLCI number
  - interface—Identifies an interface in *slot/port:channel/subchannel* format or a subinterface in *slot/port:channel/subchannel.subinterface* format
  - PVC status—Status of the circuit; valid states are *active* and *inactive.*
  - Number of circuit status inactive transitions—number of times a circuit came down because of error conditions
  - Time since creation—Time since the PVC was created
  - last status change—Time since the PVC status last changed
  - In pkts—Number of incoming packets received on the circuit
  - Out pkts—Number of outgoing packets transmitted on the circuit
  - In bytes—Number of input bytes received on the circuit
  - Out bytes—Number of output bytes received on the circuit
  - In FECN pkts—Number of packets received with the forward explicit congestion notification (FECN) bit set. The FECN bit is set by a network to notify the user that congestions may be experienced by data traffic in the direction of the frame carrying the FECN bit. The FECN bit is set by the network (not by the transmitting user), and there is no obligation for end systems to take any action regarding the FECN bit.
  - Out FECN pkts—Number of packets transmitted with the FECN bit set
  - In BECN pkts—Number of packets received with the backward explicit congestion notification (BECN) bit set. The BECN bit is set by a network to notify the user that congestions may be experienced by data traffic in the opposite direction of the frame carrying the BECN bit. The BECN bit is set by the network, and there is no obligation for end systems to take any action regarding the BECN bit.
  - Out BECN pkts—Number of packets transmitted with the BECN bit set
  - In DE pkts—Number of packets received with the discard eligibility (DE) bit set. When the DE bit is set, it indicates that the frame is discarded in preference to other frames without the DE bit set. The DE bit may be set by the network or the user. Once it is set, it cannot be reset by the user.
  - Out DE pkts—Number of packets transmitted with the DE bit set
  - Dropped packets—Number of dropped packets

- Example

```
host1#show frame-relay pvc
PVC information for frame relay NNI interface 3/2:1/1

DLCI 101 in sub-interface 3/2:1/1.1, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:29, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 0
DLCI 102 in sub-interface 3/2:1/1.2, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:28, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 0
DLCI 103 in sub-interface 3/2:1/1.3, status is active
Number of circuit status inactive transitions is 0
Time since creation 03:27:28, last status change 01:21:29
  In pkts: 0                Out pkts: 0
  In bytes: 0               Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 0
```

show frame-relay subinterface

- Use to display the state of the Frame Relay subinterface.
- The subinterface can be in one of the following states:
  - Up—Traffic can flow on the interface
  - Offline—Traffic cannot flow because hardware is unavailable
  - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
  - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
  - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- The **brief** keyword displays only the operational status of all configured subinterfaces.

- Field descriptions
  - sub-interface—Identifies the subinterface in *slot/port:channel/subchannel.subinterface* format
  - status—Status of the subinterface
  - Description—Text description or alias if configured for the subinterface
  - Time since last status change—Time since the last status change on the subinterface
  - In bytes—Number of inbound bytes received on the subinterface
  - Out bytes—Number of outbound bytes transmitted on the subinterface
  - In frames—Number of inbound frames received on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - In errors—Number of inbound errors received on the subinterface
  - Out errors—Number of outbound errors transmitted on the subinterface
  - In discards—Number of inbound packets discarded
  - Out discards—Number of outbound packets discarded
  - In unknown protos—Number of packets received on the subinterface with unknown protocols
- Example

```
host1#show frame-relay subinterface
Frame relay sub-interface 3/2:1/1.1, status is up
Description: toronto011
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0

Frame relay sub-interface 3/2:1/1.2, status is up
Description: ottawa012
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
Frame relay sub-interface 3/2:1/1.3, status is up
Description: montreal013
Time since last status change 01:21:26
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
```

show frame-relay summary

- Use to scan all defined Frame Relay interfaces and circuits; reports aggregate status as one of the following:
  - Up—Traffic can flow on the interface
  - Down—Traffic cannot flow because of a problem in the network
  - Unavailable—Traffic cannot flow because hardware is unavailable
- Example

```
host1#show frame-relay summary
28 interface(s) defined, 28 up, 0 down
840 sub-interface(s) defined, 840 up, 0 down
840 circuit(s) defined, 840 up, 0 down
```

Chapter 3

Configuring Multilink Frame Relay

This chapter describes how to configure Multilink Frame Relay (MLFR) interfaces on E-series routers.

This chapter contains the following sections:

Overview

MLFR aggregates multiple physical links into a single logical bundle. More specifically, MLFR bundles multiple link-layer channels into a single network layer channel.

The routers joined by the multilink each assign the same unique name to the bundle. A bundle can consist of multiple physical links of the same type—such as multiple asynchronous lines—or can consist of physical links of different types—such as leased synchronous lines and dial-up asynchronous lines.

The router treats MLFR like nonmultilink Frame Relay. Packets received with an MLFR header are subject to sequencing. Packets received without the MLFR header cannot be sequenced and can be delivered only on a first-come, first-served basis.

T1/E1 Connections

Some users need more bandwidth than a T1 or an E1 channel can provide, but cannot afford the expense or do not need the bandwidth of T3 or E3. Equal-cost multipath (ECMP) is one way to achieve a bandwidth greater than DS1 service without going to the expense and infrastructure required for DS3 service. MLFR is commonly used as an alternative to ECMP to deliver NxT1 service. Cost-analysis of NxT1 versus DS3 service typically imposes a practical limit of 8xT1 service; that is, aggregation of no more than eight T1 or E1 connections into an MLFR bundle.

This implementation of MLFR logically aggregates up to eight T1 or E1 connections into a single virtual connection, known as a bundle, to a given customer site. The connections can terminate at a CPE (Figure 5) or a Multilink Frame Relay bridge (Figure 6).

Figure 5: MLFR Aggregation of T1 Lines into a Single Bundle

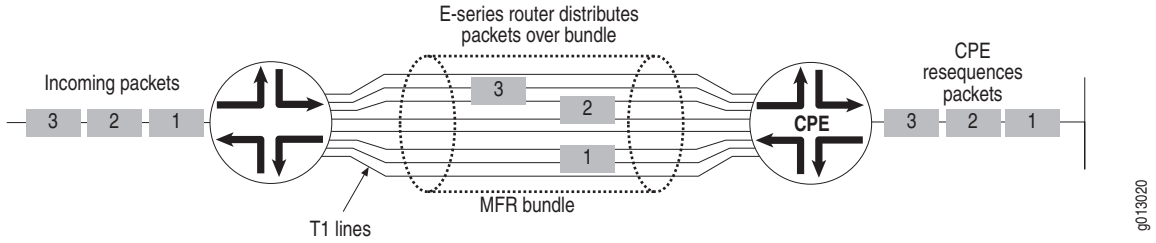
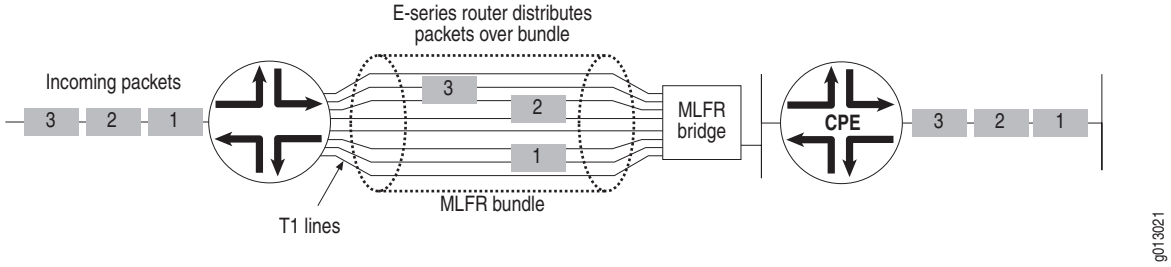


Figure 6: Terminating the Bundle at an MLFR Bridge



MLFR Link Integrity Protocol

Member links in an MLFR bundle support the MLFR Link Integrity Protocol (LIP). LIP offers several types of messages, which allow member links to join and leave a bundle. See Table 10.

Table 10: LIP Messages and Functions

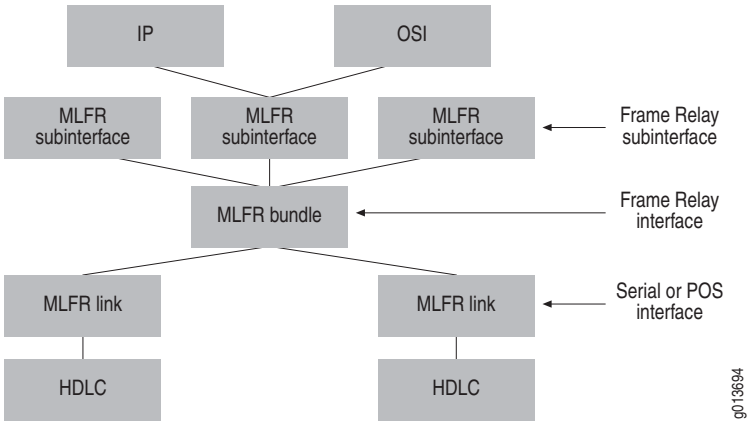
LIP Message	Function
Add-Link	Member link sends this message to request to join a bundle.
Add-Link-Ack	Member link sends this message when it receives an Add-Link message.
Add-Link-Rej	Member link sends this message to reject a request to join a bundle.
Hello	Member link sends this message to check the status of another member.
Hello-Ack	Member link sends this message when it receives a Hello message.
Remove-Link	Member link sends this message to request to leave a bundle.
Remove-Link-Ack	Member link sends this message when it receives a Remove-Link message.

The DTE creates a link management interface (LMI) with the network by encapsulating the Frame Relay frame within an MLFR frame. You assign one or more data link control identifiers (DLCIs) to a bundle.

Interface Stacking

Because MLFR aggregates multiple link-layer channels onto a single network layer IP interface, protocol layering within the router is different than it is for nonmultilink Frame Relay. See Figure 7.

Figure 7: Structure of MLFR



The MLFR Link Integrity Protocol runs on each link in a bundle. However, from the major Frame Relay interface (the bundle) upward, the interface stacking is the same as for nonmultilink Frame Relay. For example, LMI runs only on the bundle. The bundle sends and receives all MLFR packets.

Platform Considerations

You can configure MLFR interfaces on the following E-series routers:

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

NOTE: The E120 router and the E320 router do not support configuration of MLFR interfaces.

Module Requirements

For information about the modules that support MLFR interfaces on ERX-14xx models, ERX-7xx 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 MLFR.

Interface Specifiers

The interface specifier format that you use depends on the type of physical interface on which you want to configure MLFR.

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about the MLFR protocol, consult the following resources:

- Multilink Frame Relay UNI/NNI Implementation Agreement, FRF.16 (April 2000)
- Frame Relay Forum—Frame Relay Fragmentation Implementation Agreement, FRF.12 (December 1997)
- ANSI T1.617 Annex D
- ITU-T Q.933 Annex A

Supported MLFR Features

E-series routers support the following MLFR features on the cOCx/STMx and CT3 line modules:

- Logical aggregation of up to eight T1 or E1 links in a bundle
- Monotonically increasing sequence numbers for each circuit

  All packets distributed across the member links have monotonically increasing sequence numbers for each circuit. This feature enables the remote router on the customer premises to perform resequencing (if it is configured to do so).
- Static configuration of the links participating in a multilink bundle
- Round-robin packet distribution
  - On CT3 line modules, packet distribution across the member links in a bundle is handled only in a round-robin fashion. The round-robin approach is used even when the member links have different line rates.
  - On cOCx/STMx and COCX-F3 line modules, the router keeps track of the link with the least traffic. If this link cannot forward a packet, the router attempts to forward the traffic on a different link. If this attempt also fails, the router uses a round-robin approach.

You can configure bundles as follows:

- On a cOCx/STMx line module and its corresponding I/O modules, you can configure:
  - Member links from different OC3/STM1 ports in the same bundle
  - The 336 available T1 channels combined in any manner that does not exceed 8 links per bundle (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)
  - The 252 available E1 channels combined in any manner that does not exceed 8 links per bundle (for example, 252 single-link E1 bundles, 34 eight-link bundles, or 33 eight-link bundles and 8 single-link bundles)
- On a COCX-F3 line module and its corresponding I/O modules, you can configure:
  - Up to 8 member links from different ports in the same bundle
  - Up to 12 bundles

- On a CT3 or CT3/T3-F0 line module and its corresponding I/O module, you can configure:
  - Only member links from the same T3 interfaces into the same bundle. You cannot configure member links from different T3 ports in the same bundle.
  - The 28 available T1 channels on each port combined in any manner that does not exceed 8 links per bundle (for example, 28 single-link T1 bundles or 3 eight-link bundles and 4 single-link bundles per port)

Unsupported MLFR Features

E-series routers do not support the following MLFR features:

- Fragmentation

  The router does not support MLFR fragmentation or reassembly. When using MLFR on the router, configure all peer devices so that they do not fragment MLFR frames. The router drops all fragmented frames that it receives.

- Resequencing of out-of-order packets in the absence of fragmentation

  Given the location in the network where the router resides, the NxT1 links to a customer site represent one of many places across the IP network where packets might be received out of order. For example, if the router has multiple uplinks to a core router, packets might be received out of order across these links. Packet resequencing is therefore left as an exercise for the end station rather than the aggregation router.

Before You Configure MLFR

Before you begin configuring MLFR, you must configure the physical layer interfaces that will be aggregated by MLFR.

The procedures described in this chapter assume that a physical layer interface, such as a T1 or T3 interface, has been configured. For details about configuring physical layer interfaces, see the *JUNOSe Physical Layer Configuration Guide*.

Configuration Tasks

MLFR configuration consists of three major tasks, each with several steps:

1. Create the member links to be aggregated into a multilink bundle.
   a. Specify the interface on which you want to configure MLFR.

      host1(config)#**interface serial 2/0:1**

   b. Specify MLFR as the encapsulation method on the interface.

      host1(config-if)#**encapsulation mlframe-relay ietf**

2. Add member links to a multilink bundle.

   a. Define the MLFR bundle.

      host1(config)#**interface mlframe-relay boston**

   b. Add each member link.

      host1(config-if)#**member-interface serial 2/0:1**

   c. (Optional) Add a description to the major interface.

      host1(config-if)#**frame-relay description bostonBundleDescription**

   d. (Optional) Configure Frame Relay parameters.

      host1(config-if)#**frame-relay intf-type dce**
      host1(config-if)#**frame-relay lmi-type cisco**

3. Configure the Frame Relay subinterface.

   a. Define the subinterface for the MLFR bundle.

      host1(config)#**interface mlframe-relay boston.1**

   b. Assign a DLCI for the subinterface.

      host1(config-subif)#**frame-relay interface-dlci 16 ietf**

   c. (Optional) Add a description to the subinterface.

      host1(config-subif)#**frame-relay description bostonBundleSubOneDescription**

   d. Assign an IP address to the subinterface.

      host1(config-subif)#**ip address 10.10.100.1 255.255.255.0**

Configuration Example

The following commands configure three T1 lines and aggregate them into a multilink bundle named boston.

```
host1(config)#interface serial 2/0:1
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface serial 2/0:2
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface serial 2/0:3
host1(config-if)#encapsulation mlframe-relay ietf
host1(config-if)#exit
host1(config)#interface mlframe-relay boston
host1(config-if)#member-interface serial 2/0:1
host1(config-if)#frame-relay description bostonBundleDescription
host1(config-if)#frame-relay intf-type dce
host1(config-if)#frame-relay lmi-type cisco
```

```
host1(config-if)#member-interface serial 2/0:2
host1(config-if)#member-interface serial 2/0:3
host1(config-if)#exit
host1(config)#interface mlframe-relay boston.1
host1(config-subif)frame-relay description bostonBundleSubOneDescription
host1(config-subif)#frame-relay interface-dlci 16 ietf
host1(config-subif)#ip address 10.10.100.1 255.255.255.0
```

Configuring Frame Relay Versus MLFR

All the configuration commands that apply to Frame Relay also apply to MLFR. The following listing describes commands specific to configuring MLFR; for other Frame Relay commands, see *Chapter 2, Configuring Frame Relay*.

encapsulation mlframe-relay ietf

- Use to configure MLFR as the encapsulation method on an individual interface.
- Use this command only within the context of an individual interface. Issuing this command creates an MLFR link, also referred to as an MLFR bundle member.
- Example

  ```
  host1(config)#interface serial 2/0:1
  host1(config-if)#encapsulation mlframe-relay ietf
  ```

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

interface mlframe-relay

- Use to create a Frame Relay major interface, also known as the MLFR bundle.
- Example

  ```
  host1(config-if)#interface mlframe-relay group2
  ```

- Use the **no** version to delete the MLFR bundle.

member-interface

- Use to add an MLFR interface—also known as an MLFR bundle member—to an MLFR bundle.
- Example

  ```
  host1(config-if)#member-interface serial 2/0:1
  ```

- Use the **no** version to remove the specified interface from the MLFR bundle.

Monitoring MLFR

Use the commands in this section to display information about MLFR interfaces.

You can set a statistics baseline for an MLFR bundle or subinterface using the **baseline frame-relay interface mlframe-relay** command. Similarly, you can set a statistics baseline for an MLFR link with the **baseline frame-relay multilink interface** command. Use the **delta** keyword with the **show** commands described below to display statistics with the baseline values subtracted.

After you configure multilink Frame Relay, you can use the **show frame-relay** commands to view information about the multilink. For information about these commands, see *Chapter 2, Configuring Frame Relay*.

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. Refer to *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

baseline frame-relay interface

- Use to set a statistics baseline for the Frame Relay layer on MLFR bundles, Frame Relay interfaces, subinterfaces, and circuits.
- Specify the keyword **mlframe-relay** and the name of the MLFR bundle to set a baseline for the Frame Relay statistics on an MLFR bundle.
- 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.
- You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set baselines one at a time.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message "No baseline has been set" is displayed instead.
- The regular interface statistics and LMI statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines for these statistics.
- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline frame-relay interface mlframe-relay boston**

- There is no **no** version.

baseline frame-relay multilinkinterface

- Use to set a statistics baseline for MLFR links.
- 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.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message "No baseline has been set" is displayed instead.
- The regular interface statistics and LIP statistics for interfaces are subject to the same baseline timestamp. You cannot set separate baselines for these statistics.
- Use the optional **delta** keyword with Frame Relay **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline frame-relay multilinkinterface serial 3/2**

- There is no **no** version.

show frame-relay interface

- Use to display the information about the Frame Relay layer of the interface.
- Use the **brief** keyword to display the operational status of all configured interfaces.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay interface mlframe-relay—Name of the MLFR bundle
  - Status of the major Frame Relay interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of interface down transitions—Number of interfaces that have changed to a down state
  - Time since last status change—Time since the interface last changed its state
  - In bytes—Number of inbound bytes received on the interface
  - In frames—Number of inbound frames received on the interface
  - In errors—Number of inbound errors received on the interface
  - In discards—Number of inbound packets discarded

- In unknown protos—Number of packets received on the interface with unknown protocols
- Out bytes—Number of outbound bytes transmitted on the interface
- Out frames—Number of outbound frames transmitted on the interface
- Out errors—Number of outbound errors transmitted on the interface
- Out discards—Number of outbound packets discarded

- Example 1

```
host1#show frame-relay interface brief
Frame relay interface mlframe-relayTEST, status is up
```

- Example 2

```
host1#show frame-relay interface mlframe-relay TEST
Frame relay interface mlframe-relayTEST, status is up
Number of interface down transitions is 0
Time since last status change 00:01:47
Number of configured circuits: 2
  In bytes: 452              Out bytes: 198
  In frames: 19              Out frames: 11
  In errors: 0               Out errors: 0
  In discards: 8             Out discards: 0
  In unknown protos: 0
```

- Example 3

```
host1#show frame-relay interface mlframe-relay members
Frame relay interface mlframe-relay TEST is up
   Frame relay multilink member-interface 4/0:1 is up
   Frame relay multilink member-interface 4/1:1 is up
```

show frame-relay lip

- Use to display the state of MLFR Link Integrity Protocol (LIP) on an MLFR link.
- Use the **brief** keyword to display the operational status of all configured interfaces.
- Use the **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay interface—Specifier for the Frame Relay interface
  - Status of the major Frame Relay interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of interface down transitions—Number of interfaces that have changed to a down state

- Time since last status change—Time since the interface last changed its state
- Add Links sent—Number of Add Link messages sent from this interface
- Add Links received—Number of Add Link messages received on this interface
- Add Link Acknowledgments sent—Number of Add Link acknowledgments sent from this interface
- Add Link Acknowledgments received—Number of Add Link acknowledgments received on this interface
- Add Link Rejects sent—Number of Add Link Reject messages sent from this interface
- Add Link Rejects received—Number of Add Link Reject messages received on this interface
- Hellos sent—Number of Hello messages sent from this interface
- Hellos received—Number of Hello messages received on this interface
- Hello Acknowledgments sent—Number of Hello messages sent from this interface
- Hello Acknowledgments received—Number of Hello messages received on this interface
- Remove Links sent—Number of Remove Link messages sent from this interface
- Remove Links received—Number of Remove Link messages received on this interface
- Remove Link Acknowledgments sent—Number of Remove Link acknowledgments sent from this interface
- Remove Link Acknowledgments received—Number of Remove Link acknowledgments received on this interface

- Example 1

```
host1#show frame-relay lip brief
LIP information for frame relay interface 4/0:1, status is up
Number of interface down transitions is 0
Time since last status change 00:03:16

LIP information for frame relay interface 4/1:1, status is up
Number of interface down transitions is 0
Time since last status change 00:03:20
```

- Example 2

```
host1#show frame-relay lip interface serial 4/0:1
LIP information for frame relay interface 4/0:1, status is up
Number of interface down transitions is 0
Time since last status change 00:05:19
  Add Links sent: 1
  Add Links received: 1
  Add Link Acknowledgements sent: 1
  Add Link Acknowledgements received: 1
  Add Link Rejects sent: 0
  Add Link Rejects received: 0
  Hellos sent: 32
  Hellos received: 31
  Hello Acknowledgements sent: 31
  Hello Acknowledgements received: 32
  Remove Links sent: 0
  Remove Links received: 0
  Remove Link Acknowledgements sent: 0
  Remove Link Acknowledgements received: 0
```

show frame-relay lmi

- Use to display configuration and state information and statistics about the LMI.
- You can specify an interface type and location.
- Use the **brief** keyword to display abbreviated PVC information.
- Use the **delta** keyword to specify that baselined statistics are to be shown.
- DTE field descriptions
  - Frame relay DTE interface mlframe-relay—Name of the MLFR bundle
  - N391—Value of the N391 full-status polling counter
  - N392—Value of the N392 error threshold counter
  - N393—Value of the N393 monitored events counter
  - T391—Value of the T391 link integrity polling timer interval
  - Configured LMI type—One of the following options:
    - ANSI—ANSI T1.617 Annex D
    - Q933A—ITU-T Q.933 Annex A
    - Cisco—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
    - None—Suppresses LMI
  - status is up—Availability of the MLFR bundle: up or down
  - Number of interface down transitions—Number of times the interface has become unavailable
  - Time since last status change—elapsed time since LMI information changed
    - Enquiries sent—Total number of LMI status inquiries sent by the DTE on this interface
    - Full enquiries sent—Total number of LMI full status inquiries sent by the DTE on this interface

- Enquiry responses received—Total number of LMI full and regular status responses received by the DTE on this interface
- Full enquiry responses received—Total number of LMI full status responses received by the DTE on this interface
- Async updates received—Total number of asynchronous LMI updates received by the DTE on this interface
- Unknown messages received—Total number of unknown LMI messages received on this interface
- Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DTE on this interface
- No response timeouts—Total number of times a timeout occurred without receiving a response to an LMI request by the DTE on this interface
- Last sequence number sent—Last sequence number sent on this interface
- Last sequence number received—Last sequence number received on this interface

- DCE field descriptions:
  - Frame relay DCE interface mlframe-relay—Name of the MLFR bundle
  - N391—Value of the N391 full-status polling counter
  - N392—Value of the N392 error threshold counter
  - T392—Value of the T392 polling verification timer
  - Configured LMI type: one of the following options:
    - ANSI—ANSI T1.617 Annex D
    - Q933A—ITU-T Q.933 Annex A
    - Cisco—Original *Group of Four* specification developed by DEC, Northern Telecom, Stratacom, and Cisco
    - None—Suppresses LMI
  - status is up—Availability of the MLFR bundle: up or down
  - Number of interface down transitions—Number of times the interface has become unavailable
  - Time since last status change—Elapsed time since LMI information changed
    - Enquiries received—Total number of LMI status inquiries received by the DCE on this interface
    - Enquiry responses sent—Total number of LMI status responses sent by the DCE on this interface
    - Full enquiry responses sent—Total number of LMI full status responses sent by the DCE on this interface
    - Async updates sent—Total number of LMI ASYNC updates sent by the DCE on this interface

- Unknown messages received—Total number of unknown LMI messages received on this interface
- Loss of sequencing detected—Total number of times a loss of sequencing in received LMI messages was detected by the DCE on this interface
- No response timeouts—Total number of times a timeout occurred without receiving a status inquiry from the DTE on this interface
- Last sequence number sent—Last sequence number sent on this interface
- Last sequence number received—last sequence number received on this interface

- Example 1

```
host1#show frame-relay lmi brief
LMI information for frame relay DTE interface mlframe-relayTEST
DTE parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
Configured LMI type is ANSI, status is up
Number of interface down transitions is 0
Time since last status change 00:05:39
```

- Example 2

```
host1#show frame-relay lmi interface mlframe-relay TEST
LMI information for frame relay DTE interface mlframe-relayTEST
DTE parameter N391 is 6, N392 is 3, N393 is 4, T391 is 10
Configured LMI type is ANSI, status is up
Number of interface down transitions is 0
Time since last status change 00:06:20
  Enquiries sent: 39
  Full enquiries sent: 7
  Enquiry responses received: 39
  Full enquiry responses received: 7
  Async updates received: 0
  Unknown messages received: 0
  Loss of sequencing detected: 0
  No response timeouts: 0
  Last sequence number sent: 39
  Last sequence number received: 39
```

show frame-relay map

- Use to display the current Frame Relay and MLFR map entries.
- Field descriptions
  - subinterface—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - State of the subinterface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer

- LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
- AdministrativelyDown—Traffic cannot flow because of manual administrative intervention

- DLCI number—Decimal value, hexadecimal value, and value as it appears on the wire of the DLCI

- Example

```
host1#show frame-relay map
Frame relay sub-interface mlframe-relayTEST.1 (up): DLCI 16(0x10,0x4)
Frame relay sub-interface mlframe-relayTEST.2 (up): DLCI 17(0x11,0x14)
```

show frame-relay multilinkInterface

- Use to display the statistics about all MLFR interfaces or the specified MLFR interfaces.
- Field descriptions
  - Multilink Frame relay interface—Specifier for the Frame Relay interface
  - State of the MLFR interface—One of the following states:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of multilink interface down transitions—Number of interfaces that have changed to a down state
  - Time since last status change—Time since the interface last changed its state
  - In bytes—Number of inbound bytes received on the interface
  - In frames—Number of inbound frames received on the interface
  - In errors—Number of inbound errors received on the interface
  - In discards—Number of inbound packets discarded
  - In unknown protos—Number of packets received on the interface with unknown protocols
  - Out bytes—Number of outbound bytes transmitted on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - Out errors—Number of outbound errors transmitted on the interface
  - Out discards—Number of outbound packets discarded

- Example

```
host1#show frame-relay multilinkInterface
Multilink Frame relay interface 6/2:2, status is down
Number of multilink interface down transitions is 0
Time since last status change 2 days, 23 hours
  In bytes: 0                Out bytes: 0
  In frames: 0               Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 0             Out discards: 0
  In unknown protos: 0
```

show frame-relay pvc

- Use to display statistics about PVCs for Frame Relay interfaces.
- Specify a DLCI number or an interface type and location.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- The **brief** keyword displays abbreviated PVC information.
- Field descriptions
  - DLCI—DLCI number
  - subinterface—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - status—Status of the PVC
  - Number of circuit status inactive transitions—Number of times a circuit came down because of error conditions
  - Time since creation—Time since the PVC was created
  - last status change—Time since the PVC status last changed
  - In pkts—Number of incoming packets received on the circuit
  - Out pkts—Number of outgoing packets transmitted on the circuit
  - In bytes—Number of input bytes received on the circuit
  - Out bytes—Number of output bytes received on the circuit
  - In FECN pkts—Number of packets received with the forward explicit congestion notification (FECN) bit set. The FECN bit is set by a network to notify the user that data traffic may experience congestion in the direction of the frame carrying the FECN bit. The FECN bit is set by the network (not by the transmitting user), and there is no obligation for end systems to take any action regarding the FECN bit.
  - Out FECN pkts—Number of packets transmitted with the FECN bit set
  - In BECN pkts—Number of packets received with the backward explicit congestion notification (BECN) bit set. The BECN bit is set by a network to notify the user that data traffic may experience congestion in the opposite direction of the frame carrying the BECN bit. The BECN bit is set by the network, and there is no obligation for end systems to take any action regarding the BECN bit.
  - Out BECN pkts—Number of packets transmitted with the BECN bit set

- In DE pkts—Number of packets received with the discard eligibility (DE) bit set. When the DE bit is set, it indicates that the frame is discarded in preference to other frames without the DE bit set. The DE bit may be set by the network or the user. Once it is set, it cannot be reset by the user.
- Out DE pkts—Number of packets transmitted with the DE bit set
- Dropped packets—Number of dropped packets

- Example 1

```
host1#show frame-relay pvc brief
PVC information for frame relay DTE interface mlframe-relayTEST

DLCI 16 in sub-interface mlframe-relayTEST.1, status is active
DLCI 17 in sub-interface mlframe-relayTEST.2, status is active
```

- Example 2

```
host1#show frame-relay pvc interface mlframe-relay TEST
PVC information for frame relay DTE interface mlframe-relayTEST

DLCI 16 in sub-interface mlframe-relayTEST.1, status is active
Number of circuit status inactive transitions is 0
Time since creation 00:07:20, last status change 00:07:11
  In pkts: 14               Out pkts: 0
  In bytes: 420             Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 14

DLCI 17 in sub-interface mlframe-relayTEST.2, status is active
Number of circuit status inactive transitions is 0
Time since creation 00:07:20, last status change 00:07:11
  In pkts: 14               Out pkts: 0
  In bytes: 420             Out bytes: 0
  In FECN pkts: 0           Out FECN pkts: 0
  In BECN pkts: 0           Out BECN pkts: 0
  In DE pkts: 0             Out DE pkts: 0
  Dropped pkts: 14
```

show frame-relay subinterface

- Use to display the state of the subinterface.
- The subinterface can be in one of the following states:
  - Up—Traffic can flow on the interface
  - Offline—Traffic cannot flow because hardware is unavailable
  - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
  - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
  - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
- Use the **brief** keyword to display only the operational status of all configured subinterfaces.

- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - Frame relay sub-interface mlframe-relay—Name and subinterface number of the MLFR bundle in the format *bundle-name.subinterface-number*
  - status—State of the subinterface, as follows:
    - Up—Traffic can flow on the interface
    - Offline—Traffic cannot flow because hardware is unavailable
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - AdministrativelyDown—Traffic cannot flow because of manual administrative intervention
  - Number of sub-interface down transitions—Number of times a subinterface came down because of error conditions
  - Time since last status change—Time since the last status change on the subinterface
  - In bytes—Number of inbound bytes received on the subinterface
  - Out bytes—Number of outbound bytes transmitted on the subinterface
  - In frames—Number of inbound frames received on the interface
  - Out frames—Number of outbound frames transmitted on the interface
  - In errors—Number of inbound errors received on the subinterface
  - Out errors—Number of outbound errors transmitted on the subinterface
  - In discards—Number of inbound packets discarded
  - Out discards—Number of outbound packets discarded
  - In unknown protos—Number of packets received on the subinterface with unknown protocols
- Example 1

```
host1#show frame-relay subinterface brief
Frame relay sub-interface mlframe-relayTEST.1, status is up
Frame relay sub-interface mlframe-relayTEST.2, status is up
```

- Example 2

```
host1#show frame-relay subinterface mlframe-relay TEST
Frame relay sub-interface mlframe-relayTEST.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 00:07:49
  In bytes: 512              Out bytes: 0
  In frames: 16              Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 16            Out discards: 0
  In unknown protos: 0
```

```
Frame relay sub-interface mlframe-relayTEST.2, status is up
Number of sub-interface down transitions is 0
Time since last status change 00:07:50
  In bytes: 512              Out bytes: 0
  In frames: 16              Out frames: 0
  In errors: 0               Out errors: 0
  In discards: 16            Out discards: 0
  In unknown protos: 0
```

show frame-relay summary

- Use to scan all defined Frame Relay interfaces and circuits and to report the status for each discovered interface and circuit as follows:
  - Up—Traffic can flow on the interface
  - Down—Traffic cannot flow because of a problem in the network
  - Unavailable—Traffic cannot flow because hardware is unavailable
- Example

```
host1#show frame-relay summary
2 multilink interface(s) defined, 2 up, 0 down
1 interface(s) defined, 1 up, 0 down
2 sub-interface(s) defined, 2 up, 0 down
2 circuit(s) defined, 2 up, 0 down
```

Chapter 4

Configuring Upper-Layer Protocols over Static Ethernet Interfaces

This chapter describes how to configure upper-layer protocols over static Ethernet interfaces on E-series routers.

This chapter contains the following sections:

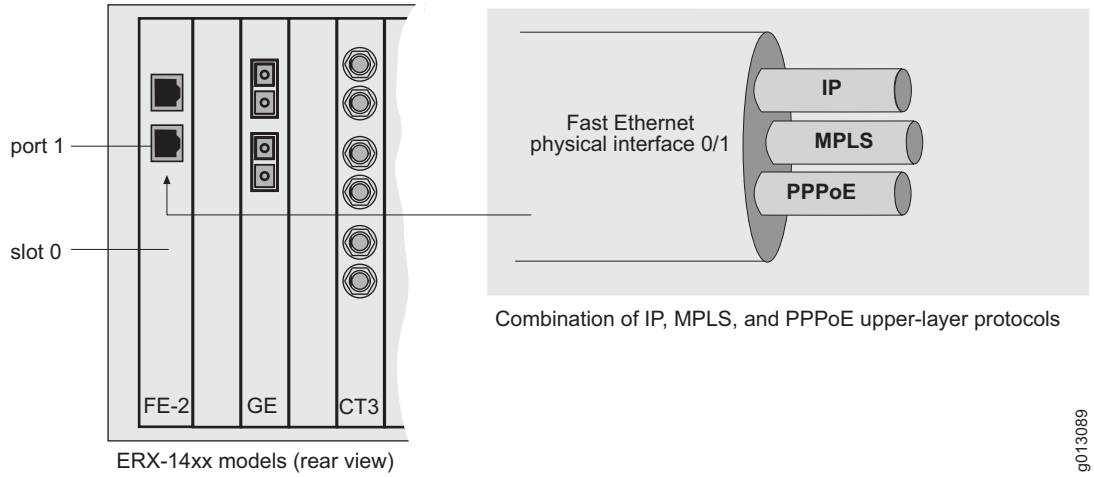
Upper-Layer Protocols over Static Ethernet Overview

You can configure one or more protocols over Ethernet with or without VLANs. This section focuses on non-VLAN configurations only. You can configure the following upper-layer protocols on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces:

- IP
- Point-to-Point Protocol over Ethernet (PPPoE)
- Multiprotocol Label Switching (MPLS)

The Ethernet configuration examples in this section use combinations of these protocols. Figure 8 on page 148 illustrates how different protocols can be multiplexed over a single physical link without the use of VLANs.

Figure 8: Multiplexing Multiple Protocols over a Single Physical Link



The following sections describe how to create the following common non-VLAN configurations, which you can configure on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces:

- IP over Ethernet
- PPPoE over Ethernet
- IP over Ethernet and MPLS over Ethernet
- IP over Ethernet, MPLS over Ethernet, and PPPoE over Ethernet

NOTE: You can also configure upper-layer protocols over dynamic interfaces. See *Chapter 15, Configuring Dynamic Interfaces* and *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Upper-Layer Protocols over Static Ethernet Platform Considerations

You can configure upper-layer protocols over Ethernet on the following E-series routers:

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

- ERX-705 router
- ERX-310 router

Module Requirements

For information about the modules supported on E-series routers:

- See the *ERX Module Guide* for modules supported on ERX-7xx models, ERX-14xx models, and the ERX-310 router.
- See the *E120 and E320 Module Guide* for modules supported on the E120 router and the E320 router.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX-7xx models, ERX-14xx models, and the ERX-310 router to specify a VLAN or S-VLAN subinterface.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies a VLAN subinterface configured on port 0 of an I/O module in slot 4.

```
host1(config)#interface fastEthernet 4/0.1
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. For example, the following command specifies a VLAN subinterface configured on port 0 of the IOA installed in the upper adapter bay of slot 3.

```
host1(config)#interface gigabitEthernet 3/0/0.1
```

For more information about interface types and specifiers on E-series models, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Upper-Layer Protocols over Static Ethernet References

For more information about upper-layer protocol implementations over Ethernet, consult the following resources:

- RFC 894—A Standard for the Transmission of IP Datagrams over Ethernet Networks (April 1984)
- RFC 1042—A Standard for the Transmission of IP Datagrams over IEEE 802 Networks (February 1988)
- RFC 1112—Host Extensions for IP Multicasting (August 1989)
- RFC 2516—Method for Transmitting PPP over Ethernet (PPPoE) (February 1998)

Configuring IP over a Static Ethernet Interface

To configure IP over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

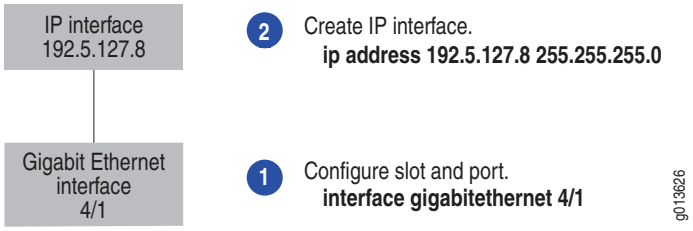
   host1(config)#**interface fastEthernet 4/1**

2. Create an IP interface.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

Figure 9 illustrates this configuration.

Figure 9: Example of IP over Ethernet Stacking Configuration Steps



Configuring PPPoE over a Static Ethernet Interface

To configure PPPoE over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

3. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1**

4. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

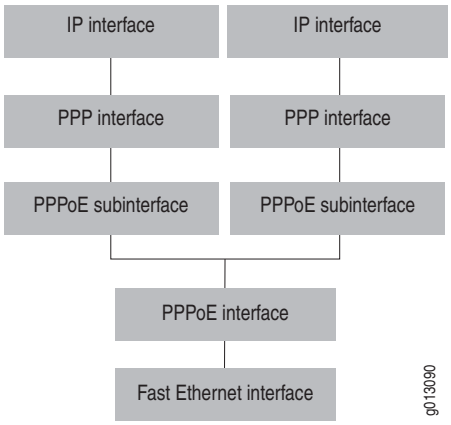
5. Assign an IP address and mask.

   host1(config-if)#**ip address 164.10.6.51 255.255.255.0**

6. (Optional) Configure additional PPPoE subinterfaces by completing Steps 3 through 5 using unique numbering.

Figure 10 illustrates this configuration.

Figure 10: Example of PPPoE Stacking Configuration Steps



Configuring IP and MPLS over a Static Ethernet Interface

To configure both IP and MPLS over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Create an IP interface.

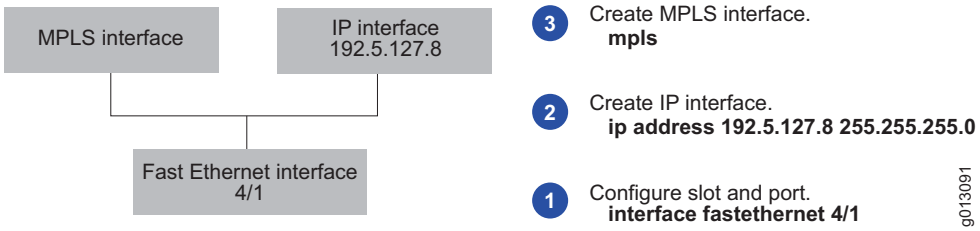
   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

3. Create an MPLS interface.

   host1(config-if)#**mpls**

Figure 11 illustrates this configuration.

Figure 11: Example of IP and MPLS Stacking Configuration Steps



Configuring IP, MPLS, and PPPoE over Ethernet

To configure IP, MPLS, and PPPoE over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Create an IP interface.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

3. Create an MPLS interface.

   host1(config-if)#**mpls**

4. Create a PPPoE interface by specifying PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

5. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1**

6. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

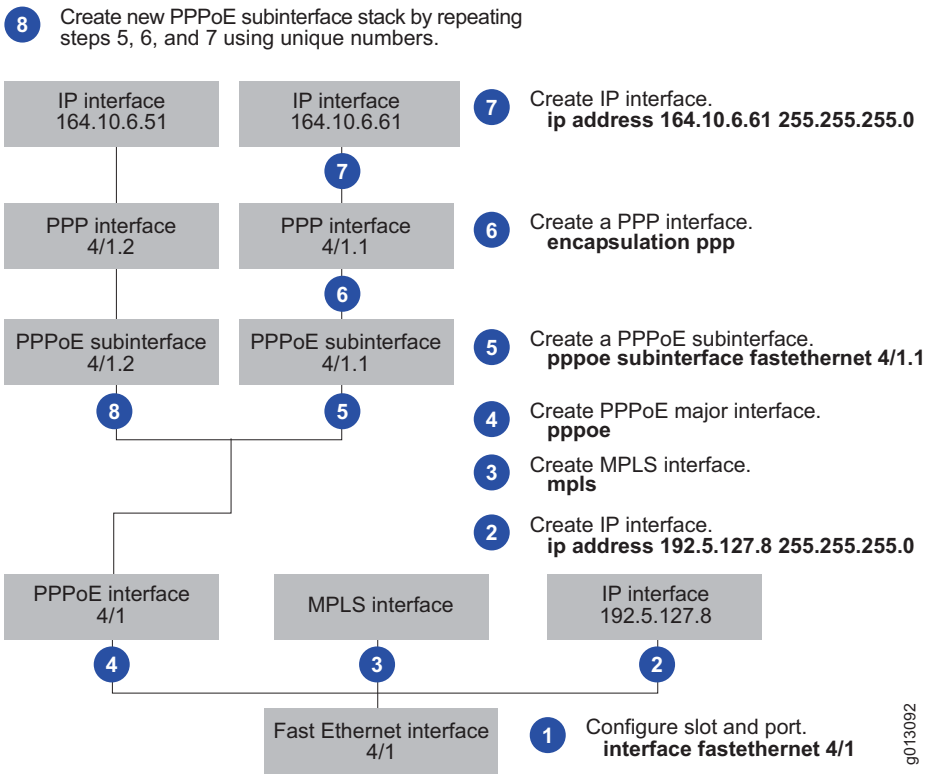
7. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

8. (Optional) Configure additional PPPoE subinterfaces by completing Steps 5 through 7 using unique numbering.

Figure 12 illustrates this configuration.

Figure 12: Example of IP, MPLS, and PPPoE Stacking Configuration Steps



mpls

- Use to enable, disable, or delete MPLS on an interface. MPLS is disabled by default.
- Example

  host1(config)#**mpls**

- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.

L2TP and Ethernet

Most Ethernet interfaces support L2TP. To use L2TP, you must first create a PPP interface. See *JUNOSe Broadband Access Configuration Guide, Chapter 11, L2TP Overview* for information about configuring L2TP.

Multinetting and Ethernet

Ethernet interfaces, except for bridged Ethernet interfaces, support multinetting; that is, adding more than one IP address to an IP interface. If you want to add multiple IP addresses to a single IP interface, use the **ip address** command with the **secondary** keyword, which is described in *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*.

Monitoring Upper-Level Protocols over Ethernet

This section explains how to use the **show** commands to display the physical characteristics and the configured settings for Ethernet interfaces.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

You can use various **show** commands to monitor upper-layer protocols. For more information, see:

- *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*
- *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*
- *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS*

show interfaces fastEthernet

- Use to display the status of Fast Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions
  - FastEthernet *interfaceSpecifier*—Status of the hardware on this interface
    - up—Hardware is operational
    - down—Hardware is not operational
  - Administrative status—Operational state that you configured for this interface
    - up—Interface is enabled
    - down—Interface is disabled
  - Hardware—Type of MAC device on this interface
  - Address—MAC address of the processor on this interface

- MAU—Type of medium attachment unit (MAU) on the physical port:
  - 10BASE-T (10 Mbps)
  - 100BASE-TX (100 Mbps)
  - 100BASE-FX-MM (100 Mbps) with the distance appearing after the type
  - 100BASE-LX-SM (100 Mbps)
  - SFP (Empty)—SFPs that are empty
  - SFP (Non-compliant Juniper Part)—SFPs that are installed in the FE-8 I/O module and do not have a Juniper Networks part number programmed
- MTU—Size of the MTU for this interface
  - Operational—Size of the largest packet processed
  - Administrative—Setting for MTU size that you specified
- Duplex Mode—Duplex option for this interface
  - Operational—Duplex option currently used
  - Administrative—Setting for duplex that you specified
- Speed—Line speed for this interface
  - Operational—Current rate at which packets are processed
  - Administrative—Setting for line speed
  - 5 minute input rate—Data rates based on traffic received in the last 5 minutes
  - 5 minute output rate—Data rates based on traffic sent in the last 5 minutes
- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received in error-free packets
  - Unicast—Number of unicast packets received
  - Multicast—Number of multicast packets received
  - Broadcast—Number of broadcast packets received
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
  - Mac Errors—Number of incoming packets discarded because of MAC sublayer failures
  - Alignment—Number of incomplete octets received
  - CRC—Number of packets discarded because the checksum the router computed from the data does not match the checksum generated by the originating devices
  - Too Longs—Number of packets discarded because the size exceeded the MTU
  - Symbol Errors—Number of symbols received that the router did not correctly decode

- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent
  - Unicast—Number of unicast packets sent
  - Multicast—Number of multicast packets sent
  - Broadcast—Number of broadcast packets sent
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
  - Mac Errors—Number of outgoing packets discarded because of MAC sublayer failures
  - Deferred—Number of packets that the router delayed sending because the line was busy. In half duplex mode, a high number of deferrals means the link is very busy with traffic from other stations. In full duplex mode, when the link is always available for transmission, this number is zero.
  - No Carrier—Number of packets sent when carrier sense was unavailable
- Collisions—Analysis of the collisions that occurred
  - Single—Number of packets sent after one collision
  - Multiple—Number of packets sent after multiple collisions
  - Late—Number of packets aborted during sending because of collisions after 64 bytes
  - Excessive—Number of packets not sent because of too many collisions
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, bytes—Number of packets and bytes that were forwarded on this queue
  - Dropped committed packets, bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, bytes—Number of exceeded packets and bytes that were dropped

- Example—Displays the status of a Fast Ethernet interface

```
host1:vr2#show interfaces fastEthernet 2/0
FastEthernet2/0 is Up, Administrative status is Up
  Hardware is Intel 21440, address is 0090.1a10.0552
  MAU is 10BASE-T
  MTU: Operational 1518, Administrative 1518
  Duplex Mode: Operational Full Duplex, Administrative Auto Negotiate
  Speed: Operational 100 Mbps, Administrative Auto Negotiate

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

  In: Bytes 39256, Unicast 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0, Mac Errors 0, Alignment 0
   CRC 0, Too Longs 0, Symbol Errors 0
  Out: Bytes 4579036, Unicast 610
   Multicast 0, Broadcast 70932
   Errors 0, Discards 0, Mac Errors 0, Deferred 0, No Carrier 3
   Collisions: Single 0, Multiple 0, Late 0, Excessive 0
 ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
Administrative qos-shaping-mode: none
Operational qos-shaping-mode: none

queue 0: traffic class control, bound to FastEthernet2/0
  Queue length 0 bytes
  Forwarded packets 1, bytes 46
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0
```

show interfaces gigabitEthernet
show interfaces tenGigabitEthernet

- Use to display the status of Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions
  - GigabitEthernet or tenGigabitEthernet *interfaceSpecifier*—Status of the hardware on this interface
    - up—Hardware is operational
    - down—Hardware is not operational
  - Administrative status—Operational state that you configured for this interface
    - up—Interface is enabled
    - down—Interface is disabled

- Hardware—Type of MAC device on this interface
- Address—MAC address of the processor on this interface
- MAU—Type of medium attachment unit (MAU) on the primary and secondary physical ports:
  - SFP—1000BASE-LH, 1000BASE-SX, 1000BASE-ZX; for SFPs that are empty, SFP (Empty) appears in this field; for SFPs that are installed in the OC3-2 GE APS I/O module and do not have a Juniper Networks part number programmed, SFP (GE Compliant) appears in this field
  - XFP—10GBASE-SR (10 Gbps), 10GBASE-LR (10 Gbps), 10GBASE-ER (10 Gbps); for XFPs that are empty, XFP (Empty) appears in this field
- MTU—Size of the MTU for this interface
  - Operational—Size of the largest packet processed
  - Administrative—Setting for MTU size that you specified
- Duplex Mode—Duplex option for this interface
  - Operational—Duplex option currently used
  - Administrative—Setting for duplex that you specified
- Speed—Line speed for this interface
  - Operational—Current rate at which packets are processed
  - Administrative—Setting for line speed that you specified
- Link—Link information for this interface
  - Operational Link Selected—Port that the I/O module is currently using: primary or secondary
  - Administrative link selected—Port that the I/O module is configured to use:
    - primary—Only primary port is configured to operate
    - secondary—Only redundant port is configured to operate
    - automatically—Software controls port redundancy automatically
- Primary link selected x times—Number of times that the I/O has used the primary port since the module was last rebooted
- Secondary link selected x times—Number of times that the I/O has used the secondary port since the module was last rebooted
- Primary/Secondary link signal detected, Primary/Secondary link signal not detected—Specifies the port (primary or secondary) on which the router detects a signal
- 5 minute input rate—Data rates based on the traffic received in the last 5 minutes
- 5 minute output rate—Data rates based on the traffic sent in the last 5 minutes

- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received in error-free packets
  - Unicast—Number of unicast packets received
  - Multicast—Number of multicast packets received
  - Broadcast—Number of broadcast packets received
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
  - Mac Errors—Number of incoming packets discarded because of MAC sublayer failures
  - Alignment—Number of incomplete octets received
  - CRC—Number of packets discarded because the checksum that the router computed from the data does not match the checksum generated by the originating devices
  - Too Longs—Number of packets discarded because the size exceeded the MTU
  - Symbol Errors—Number of symbols received that the router did not correctly decode
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent
  - Unicast—Number of unicast packets sent
  - Multicast—Number of multicast packets sent
  - Broadcast—Number of broadcast packets sent
  - Errors—Total number of errors in all transmitted packets; note that some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
  - Mac Errors—Number of outgoing packets discarded because of MAC sublayer failures
  - Deferred—Number of packets that the router delayed sending because the line was busy. In half duplex mode, a high number of deferrals means the link is very busy with traffic from other stations. In full duplex mode, when the link is always available for transmission, this number is zero.
  - No Carrier—Number of packets sent when carrier sense was unavailable
- Collisions—Analysis of the collisions that occurred
  - Single—Number of packets sent after one collision
  - Multiple—Number of packets sent after multiple collisions
  - Late—Number of packets aborted during sending because of collisions after 64 bytes
  - Excessive—Number of packets not sent because of too many collisions

- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, bytes—Number of packets and bytes that were forwarded on this queue
  - Dropped committed packets, bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, bytes—Number of exceeded packets and bytes that were dropped

- Example—Displays the status of a Gigabit Ethernet interface

```
host1:vr2#show interfaces gigabitEthernet 10/2
ERX-40-20-43#show int gigabitEthernet 10/2
GigabitEthernet10/2 is Down, Administrative status is Up
  Hardware is SEEQ 8101, address is 0090.1a01.0cc8
  Primary MAU is 1000BASE-SX, secondary MAU is SFP (Empty)
  MTU: Operational 1518, Administrative 1518
  Duplex Mode: Operational Full Duplex, Administrative Auto Negotiate
  Speed: Operational 1000 Mbps, Administrative Auto Negotiate
  Link: Operational Secondary Link Selected,
        Administrative Link Selected Automatically
  Link Failover Timeout: Operational 652 ms, Administrative default
  Primary link selected 6302 times, Secondary link selected 6302 times
  Primary link signal detected, Secondary link signal detected

  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

  In: Bytes 0, Unicast 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0, Mac Errors 0, Alignment 0
   CRC 0, Too Longs 0, Symbol Errors 0
  Out: Bytes 0, Unicast 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0, Mac Errors 0, Deferred 0, No Carrier 0
   Collisions: Single 0, Multiple 0, Late 0, Excessive 0
 ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
Administrative qos-shaping-mode: none
Operational qos-shaping-mode: none
```

```
queue 0: traffic class control, bound to GigabitEthernet10/2
  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
```

Chapter 5

Configuring VLAN and S-VLAN Subinterfaces

This chapter describes how to configure VLAN and S-VLAN subinterfaces on E-series routers.

This chapter contains the following sections:

VLAN Overview

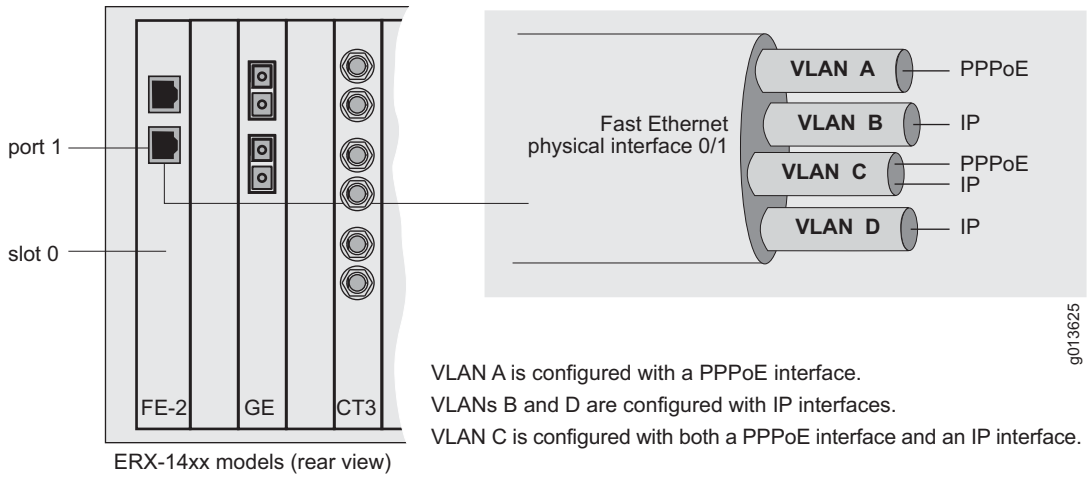
A virtual LAN (VLAN) enables multiplexing multiple IP and PPPoE interfaces and MPLS interfaces over a single physical Ethernet port. This multiplexing is accomplished through VLAN subinterfaces. Ethernet interfaces support the 802.1q-1998 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, which the router uses as its standardized format for frame tagging.

The Ethernet V2 frame format enables multiplexing of different protocols over a single physical link. IEEE 802.1q compatibility extends the frame format by adding a tag that contains a VLAN ID. This feature enables multiplexing of different channels (VLANs) over the physical link; each channel is able to multiplex different protocols.

This capability works very much like ATM encapsulation as described in RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999). This encapsulation type enables multiplexing of multiple protocols over a single ATM virtual circuit (VC).

As shown in Figure 13, VLANs are similar to ATM VCs, with the VLAN ID serving the same function as the virtual path identifier (VPI) and virtual channel identifier (VCI) to multiplex the different channels over the physical link. The Ethernet protocol type serves the same function within a VLAN as the logical link control (LLC) subnetwork attachment point (SNAP) within a VC, to multiplex the different protocols over the channel.

Figure 13: Use of VLANs to Multiplex Different Protocols over a Single Physical Link



In a VLAN configuration, the router can send VLAN 0 *tagged* or *untagged frames*.

All VLAN subinterfaces use the MAC address of the Ethernet interface over which they are configured. However, some configurations, such as multiple IP over VLAN subinterfaces, require that you connect many VLAN subinterfaces to a single device. In these cases, the device uses the MAC address to identify and select the correct VLAN to use. When the MAC address is the same for all VLANs, uneven load balancing of traffic occurs. To ensure proper load balancing, you must assign unique MAC addresses to the individual VLAN subinterfaces that are connected to the device. Any ARP requests and responses generated for the IP address assigned to a VLAN subinterface use this MAC address.

You must assign the MAC address when you configure the VLAN ID. If you change the MAC address of the VLAN subinterface after you configure it, system errors can occur. To change the MAC address, you must first remove the VLAN subinterface and then reconfigure it.

For more information, see:

- *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*
- *JUNOSe Link Layer Configuration Guide, Chapter 10, Configuring Point-to-Point Protocol over Ethernet*

S-VLAN Overview

As described in *VLAN Overview* on page 163, VLANs permit multiplexing multiple IP interfaces and PPPoE interfaces over a single physical Ethernet port by creating VLAN subinterfaces. As specified in IEEE Standard 802.1q, the 12-bit VLAN identifier's tagged frames enables the construction of a maximum of 4096 distinct VLANs. In an Ethernet B-RAS application environment, however, this VLAN limit is inadequate. A stacked VLAN (S-VLAN) provides a two-level VLAN tag structure, which extends the VLAN ID space to more than 16 million VLANs.

Creating an S-VLAN requires the use of a second encapsulation tag. The router performs decapsulation twice, once to get the S-VLAN tag and once to get the VLAN tag. This *double tagging* approach enables more than 16 million address possibilities, which more than satisfies the scaling requirement for Ethernet B-RAS applications.

VLAN and S-VLAN subinterfaces can coexist over the same VLAN major interface. You configure S-VLANs in the same way that you configure VLANs, with the addition of certain commands.

NOTE: See *JUNOSe Release Notes*, *Appendix A*, *System Maximums* for S-VLAN limitations.

Like VLANs, all S-VLAN subinterfaces use the MAC address of the Ethernet interface over which they are configured. For more information about assigning unique MAC address to the S-VLAN subinterface when assigning VLAN or S-VLAN IDs, see *VLAN Overview* on page 163.

VLAN and S-VLAN Platform Considerations

You can configure VLAN and S-VLAN subinterfaces on the following E-series routers:

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

Module Requirements

For information about the modules supported on E-series routers:

- See the *ERX Module Guide* for modules supported on ERX-7xx models, ERX-14xx models, and the ERX-310 router.
- See the *E120 and E320 Module Guide* for modules supported on the E120 router and the E320 router.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX-7xx models, ERX-14xx models, and the ERX-310 router to specify a VLAN or S-VLAN subinterface.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies a VLAN subinterface configured on port 0 of an I/O module in slot 4.

```
host1(config)#interface fastEthernet 4/0.1
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. For example, the following command specifies a VLAN subinterface configured on port 0 of the IOA installed in the upper adapter bay of slot 3.

```
host1(config)#interface gigabitEthernet 3/0/0.1
```

For more information about interface types and specifiers on E-series models, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

VLAN and S-VLAN References

For more information about VLAN and S-VLAN implementations, consult the following resources:

- IEEE 802.1q (Virtual LANs)

Creating a VLAN Subinterface

Ethernet interfaces support IP, PPPoE, MPLS, or both IP and PPPoE on each VLAN. In addition to a VLAN major interface level, a VLAN subinterface level distinguishes the VLAN.

NOTE: You cannot configure VLANs on the Fast Ethernet port of the SRP module.

Tasks to configure VLAN subinterface are:

Creating a VLAN Major Interface

To use VLANs, you must first configure the Ethernet interface for VLAN encapsulation. This creates the VLAN major interface. For example:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The router creates the VLAN major interface.

You can now create multiple VLAN subinterfaces to carry higher-level protocols. For examples, see *Creating a VLAN Subinterface*, next.

Configuring IP over VLAN

To configure IP over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/0.3**

4. Do one of the following:

   a. Assign a VLAN ID for the subinterface.

      host1(config-if)#**vlan id 201**

   b. Assign a VLAN ID and the optional unique MAC address for the subinterface.

      host1(config-if)#**vlan id 201 mac-address 0090.1a01.1234**

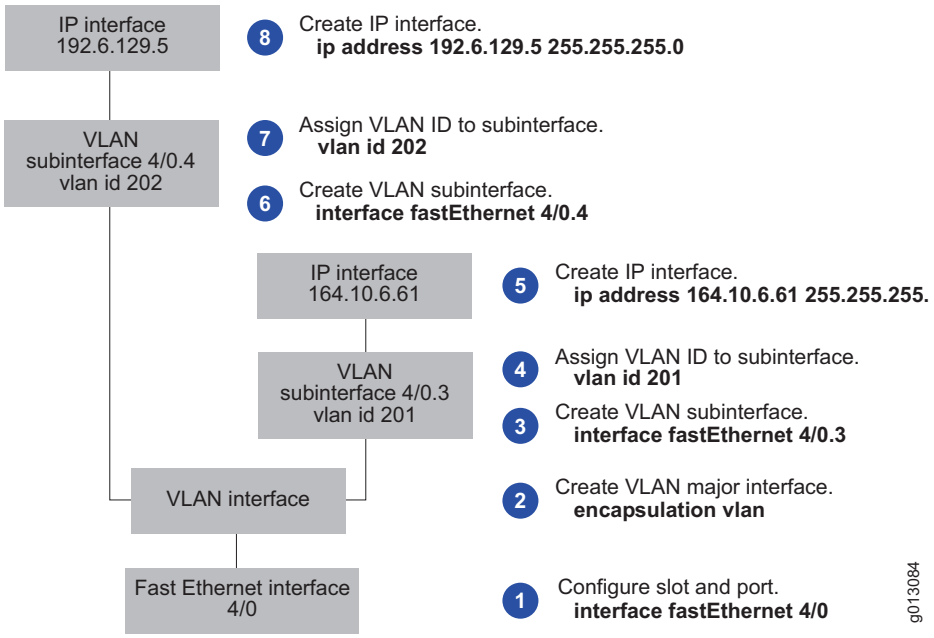
5. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

6. (Optional) Configure additional VLAN subinterfaces by completing Steps 3 through 5.

Figure 14 illustrates the IP/VLAN/Fast Ethernet stacking, showing two separate VLAN subinterfaces. Configure one VLAN subinterface entirely; then configure the next VLAN subinterface.

Figure 14: Example of IP/VLAN/Fast Ethernet Stacking Configuration Steps



Configuring PPPoE over VLAN

To configure PPPoE over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 201**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#**vlan id 201 mac-address 0090.1a01.1234**

5. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

6. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

7. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

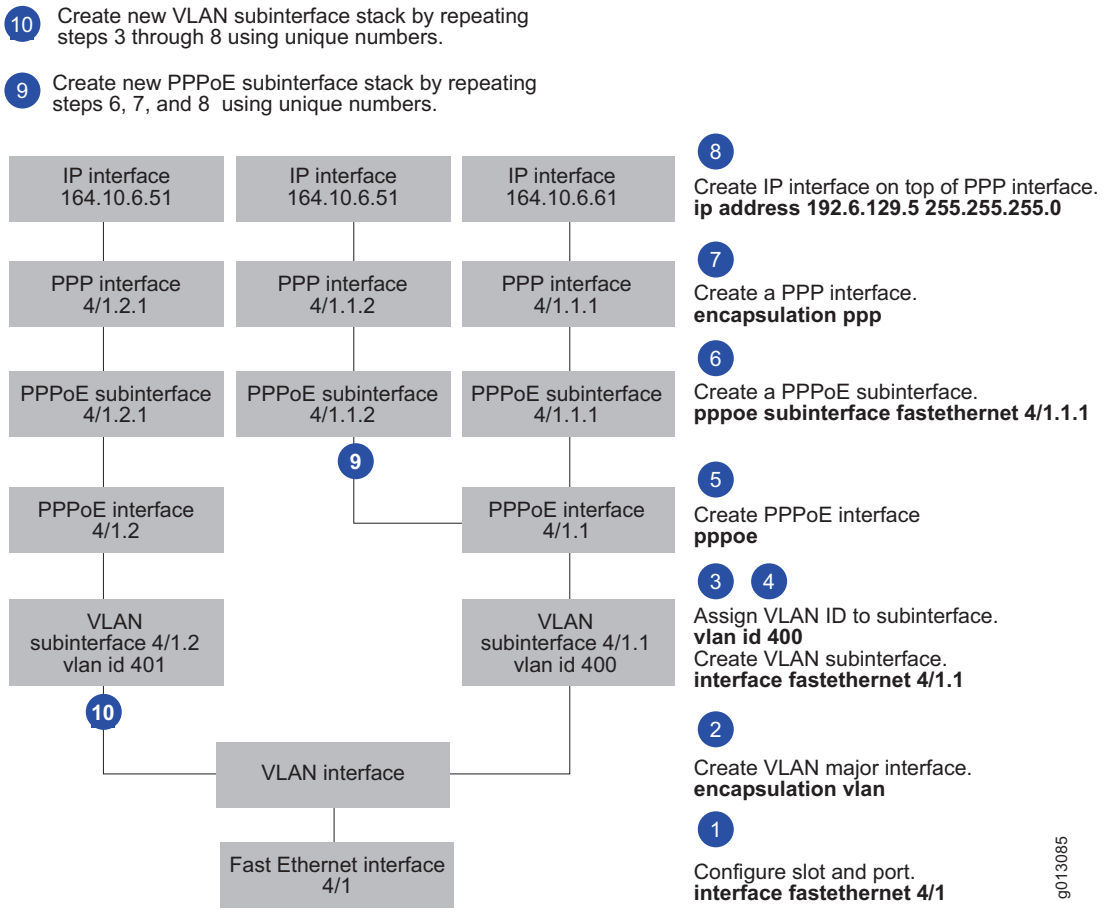
8. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

9. (Optional) Configure additional VLAN subinterfaces by completing Steps 3 through 8.

Figure 15 illustrates the PPPoE/VLAN/Fast Ethernet stacking, showing two separate VLAN subinterfaces. One VLAN subinterface has two PPPoE subinterfaces, and one VLAN subinterface has one PPPoE subinterface.

Figure 15: Example of PPPoE/VLAN/Fast Ethernet Stacking Configuration Steps



Configuring MPLS over VLAN

To configure MPLS over VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 400**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

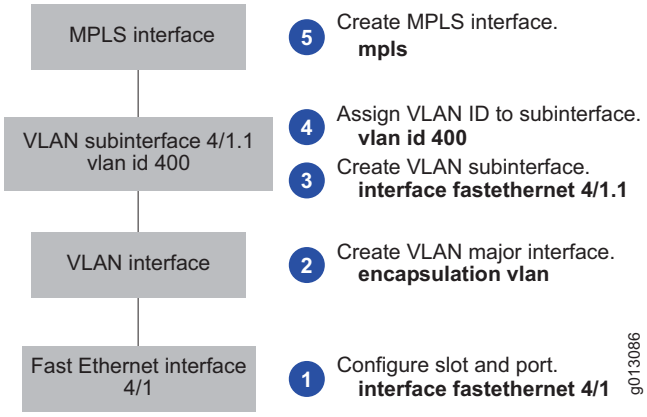
     host1(config-if)#**vlan id 400 mac-address 0090.1a01.1234**

5. Enable MPLS on the interface.

   host1(config-if)#**mpls**

Figure 16 illustrates the MPLS/VLAN/Fast Ethernet stacking, showing one VLAN subinterface.

Figure 16: Example of MPLS/VLAN/Fast Ethernet Stacking Configuration Steps



Configuring IP over VLAN and PPPoE over VLAN

To configure IP over VLAN with PPPoE over the same VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 400**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#**vlan id 400 mac-address 0090.1a01.1234**

5. Create an IP interface on the same VLAN as the PPPoE interface.

   host1(config-if)#**ip address 164.10.6.71 255.255.255.0**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

7. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

8. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

9. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

10. (Optional) Configure additional PPPoE subinterfaces by completing Steps 7 through 9 using unique numbering.

To configure additional IP interfaces over the VLAN major interface:

1. Create a new VLAN subinterface by adding a unique subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.2**

2. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 401**

3. Assign an IP address and mask.

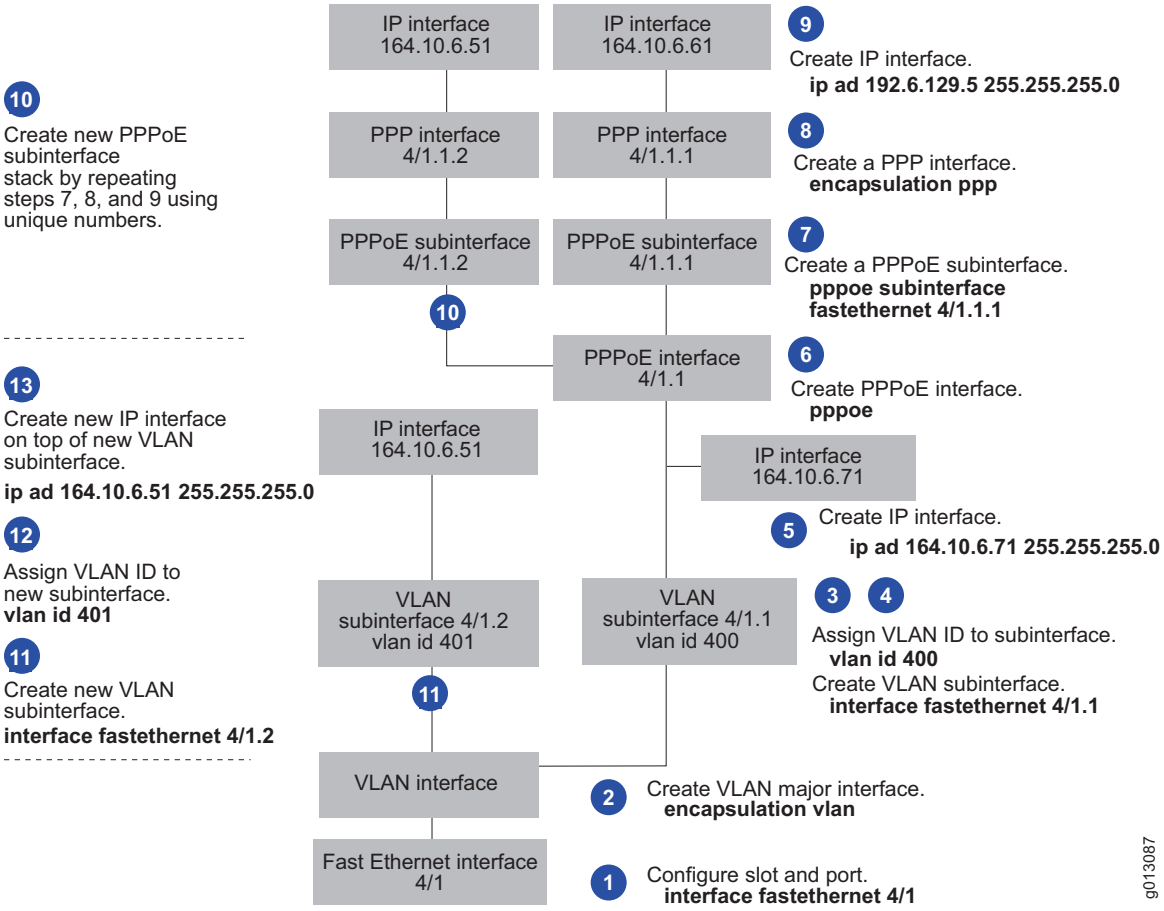
   host1(config-if)#**ip address 164.10.6.51 255.255.255.0**

Figure 17 illustrates the configuration steps for two VLAN subinterfaces. In this example:

- VLAN subinterface 4/1.1 has an IP interface, a PPPoE interface, and multiple PPPoE subinterface stacks.
- VLAN subinterface 4/1.2 has only an IP interface.

NOTE: Before you can remove a VLAN subinterface, you must remove the upper-layer interface stack.

Figure 17: Example of PPPoE over VLAN with IP over VLAN Stacking Configuration Steps



encapsulation ppp

- Use to configure PPP as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation ppp**

- Use the **no** version to disable PPP on the interface.

encapsulation vlan

- Use to configure VLAN as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation vlan**

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

ip address

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

  host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

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

pppoe

- Use to configure PPPoE as the encapsulation method on the interface.
- Example

  host1(config-if)#**pppoe**

- Use the **no** version to disable PPPoE on the interface.

pppoe subinterface fastEthernet

- Use to create a PPPoE subinterface on a Fast Ethernet interface.
- Example

  host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

- Use the **no** version to remove a PPPoE subinterface on a Fast Ethernet interface.

pppoe subinterface gigabitEthernet
pppoe subinterface tenGigabitEthernet

- Use to create a PPPoE subinterface on a Gigabit Ethernet interface or on a 10-Gigabit Ethernet interface.
- Example 1—Creates a PPPoE subinterface on an ERX-7xx model, ERX-14xx model, or the ERX-310 router

  host1(config-if)#**pppoe subinterface gigabitEthernet 4/2.1.1**

- Example 2—Creates a PPPoE subinterface on the E320 router

  host1(config-if)#**pppoe subinterface tenGigabitEthernet 4/0/2.1.1**

- Use the **no** version to remove a PPPoE subinterface on a Gigabit Ethernet interface or on a 10-Gigabit Ethernet interface.

vlan description

- Use to assign an alias or description to a VLAN subinterface.
- You can use a maximum of 64 characters for the description or to name the alias.
- Example

  host1(config-if)#**vlan description randolph56a**

- Use the **no** version to remove the VLAN description.

vlan id

- Use to specify the VLAN ID.
- Use a VLAN ID that is in the range 0–4095 and is unique within the Ethernet interface.
- Issue the **vlan id** command before any upper bindings are made, such as IP or PPPoE.
- Use the **mac-address** keyword to specify a unique MAC address for the VLAN subinterface. When you do not specify a unique MAC address, the VLAN uses the MAC address of the Ethernet interface.
- Use the optional keyword **untagged** to specify that frames be sent untagged. The keyword is valid only for VLAN ID 0. Tagged frames can be received, but untagged frames are sent.
- Examples

  host1(config-if)#**vlan id 400**
  host1(config-if)#**vlan id 4 255 mac-address 0090.1a01.1234**

- There is no **no** version.

Configuring a S-VLAN Subinterface

Tasks to configure a S-VLAN subinterface include:

- Configuring an S-VLAN Subinterface on page 175
- Configuring PPPoE over an S-VLAN on page 176

Configuring an S-VLAN Subinterface

To configure an S-VLAN subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 4 255**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 88a8**

Configuring PPPoE over an S-VLAN

To configure PPPoE over an S-VLAN over an Ethernet interface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   host1(config)#**interface fastEthernet 4/0**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   The VLAN major interface is added.

3. Create a VLAN subinterface by adding a subinterface number to the interface identification command.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 4 255**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 88a8**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

7. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1.1**

8. Specify PPP as the encapsulation method on the interface.

   host1(config-if)#**encapsulation ppp**

9. Assign an IP address and mask.

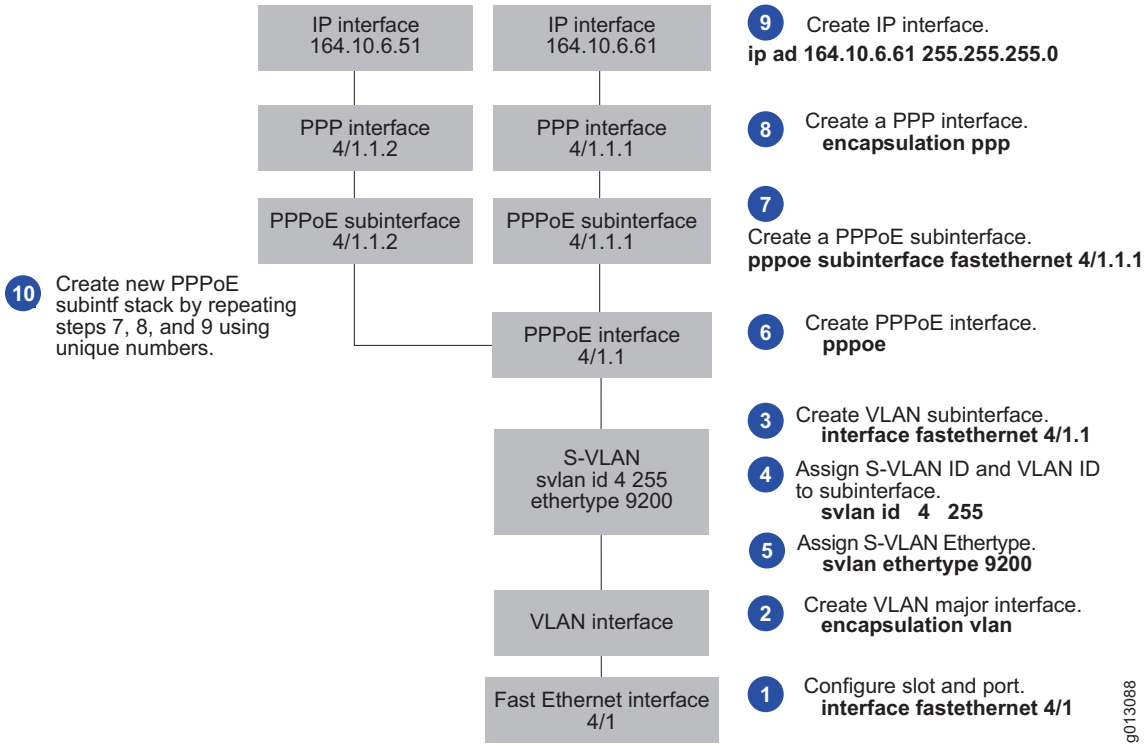
   host1(config-if)#**ip address 164.10.6.61 255.255.255.0**

10. (Optional) Configure additional PPPoE subinterfaces by completing Steps 7 through 9 using unique numbering.

Figure 18 shows one S-VLAN subinterface with multiple PPPoE subinterface stacks.

NOTE: Before you can remove an S-VLAN/VLAN subinterface, you must remove the upper-layer interface stack.

Figure 18: Example of PPPoE over S-VLAN Stacking Configuration Steps



encapsulation ppp

- Use to configure PPP as the encapsulation method for the interface.
- Use the **no** version to remove PPP as the encapsulation method on the interface.

encapsulation vlan

- Use to configure VLAN as the encapsulation method for the interface.
- Use the **no** version to remove VLAN as the encapsulation method on the interface.

ip address

- Use to set a primary or secondary IP address for an interface or subinterface.
- Specify the layer 2 encapsulation before you set the IP address.
- Use the **no** version to remove an IP address or disable IP processing.

pppoe

- Use to configure PPPoE as the encapsulation method on the interface.
- Use the **no** version to disable PPPoE on the interface.

pppoe subinterface fastEthernet

- Use to create a PPPoE subinterface on a Fast Ethernet interface.
- Use the **no** version to remove a PPPoE subinterface on a Fast Ethernet interface.

pppoe subinterface gigabitEthernet
pppoe subinterface tenGigabitEthernet

- Use to create a PPPoE subinterface on a Gigabit Ethernet interface or on a 10-Gigabit Ethernet interface.
- Use the **no** version to remove a PPPoE subinterface on a Gigabit Ethernet interface or on a 10-Gigabit Ethernet interface.

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN subinterface.
- Choose one of the following Ethertype values:
  - 8100—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - 88a8—Specifies Ethertype value 0x88a8, as defined in draft IEEE Standard 802.1ad
  - 9100—Specifies Ethertype value 0x9100, which is the default
- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-if)#**svlan ethertype 8100**

- Use the **no** version to restore the default value, 9100.

svlan id

- Use to assign S-VLAN IDs and VLAN IDs to VLAN subinterfaces.
- Use S-VLAN ID and VLAN ID numbers that are in the range 0–4095 and that are unique within the Ethernet interface.
- Use the **mac-address** keyword to specify a unique MAC address for the VLAN subinterface. When you do not specify a unique MAC address, the VLAN uses the MAC address of the Ethernet interface.

- Examples

  host1(config-if)#**svlan id 4 255**
  host1(config-if)#**svlan id 4 255 mac-address 0090.1a01.1234**

- Issue the **svlan id** command before any upper bindings are made, such as IP or PPPoE.
- There is no **no** version.

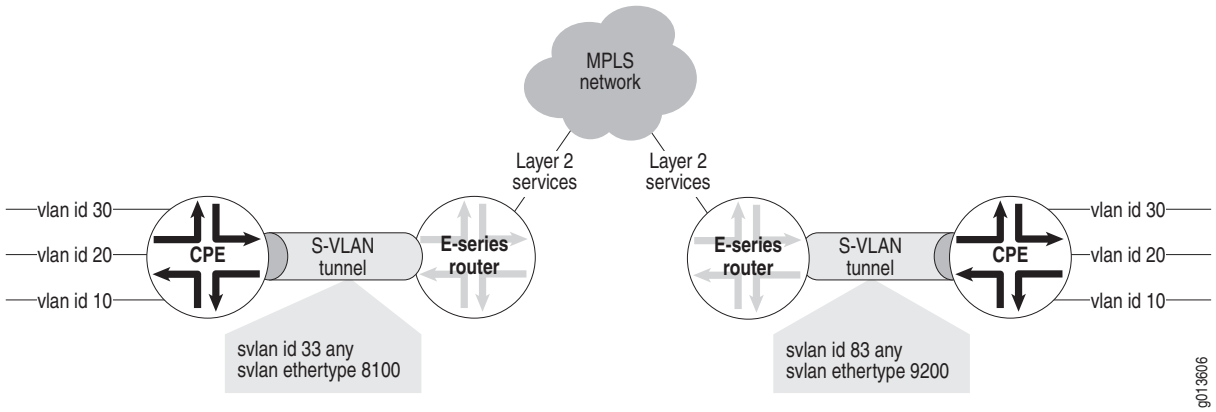
Configuring S-VLAN Tunnels for Layer 2 Services over MPLS

When you configure Ethernet layer 2 services over MPLS, you can create a special type of S-VLAN called an S-VLAN tunnel that uses a single interface to tunnel traffic from multiple VLANs across an MPLS network. The S-VLAN tunnel enables multiple VLANs, each configured with a unique VLAN ID tag, to share a common S-VLAN ID tag when they traverse an MPLS network.

Advantages

Using S-VLAN tunnels provides an easier and faster way to configure Ethernet layer 2 services over MPLS than using standard S-VLANs. For example, consider the network configuration shown in Figure 19.

Figure 19: S-VLAN Tunnels for Ethernet Layer 2 Services over MPLS



In this example, traffic from three VLAN subinterfaces must traverse the MPLS network. To accomplish this using standard S-VLANs, you issue the following commands to configure three separate S-VLANs with the same S-VLAN ID value and different VLAN IDs, as follows:

host1(config-if)#**svlan id 33 10**
host1(config-if)#**svlan id 33 20**
host1(config-if)#**svlan id 33 30**

By contrast, using an S-VLAN tunnel achieves the same result, but requires you to issue only a single **svlan id** command with the keyword **any** in place of the VLAN ID value. For example, the following command creates a single interface that tunnels traffic from VLANs configured with an S-VLAN ID of 33 and *any* VLAN ID to the same destination across the MPLS network. In effect, this command tunnels traffic from all three VLANs shown in Figure 19 on page 179.

```
host1(config-if)#svlan id 33 any
```

Interface Stacking

When you configure Ethernet layer 2 services over MPLS using S-VLAN tunnels, the only interface that you can stack over an S-VLAN tunnel is an MPLS tunnel, which you configure using the MPLS tunneling command (**mpls-relay** or **route interface**) that is appropriate for your configuration. Attempting to configure any other interface type—such as IP, MPLS (nontunnel), or PPPoE—over the S-VLAN tunnel causes the router to generate an error and reject the configuration as invalid.

For details about configuring MPLS and layer 2 services over MPLS, see:

- *JUNOSe BGP and MPLS Configuration Guide, Chapter 2, Configuring MPLS*
- *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*

Configuration Example

This section uses the sample network topology shown in Figure 19 on page 179 to illustrate the steps for configuring S-VLAN tunnels for Ethernet layer 2 services over MPLS.

To configure S-VLAN tunnels for Ethernet layer 2 services over MPLS:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet port.

   ```
   host1(config)#interface fastEthernet 4/0
   ```

2. Specify VLAN as the encapsulation method to create the VLAN major interface.

   ```
   host1(config-if)#encapsulation vlan
   ```

3. Create a VLAN subinterface.

   ```
   host1(config-if)#interface fastEthernet 8/1.1
   ```

4. Create the S-VLAN tunnel. This interface tunnels traffic from VLANs configured with an S-VLAN ID of 33 and any VLAN ID to the same destination across the MPLS network.

   ```
   host1(config-if)#svlan id 33 any
   ```

5. Assign an S-VLAN Ethertype.

   ```
   host1(config-if)#svlan ethertype 8100
   ```

6. Create the MPLS tunnel interface using the appropriate MPLS tunneling command for your configuration. For example:

   host1(config-if)#**route interface tunnel mpls:tunnel3 45**

   For complete instructions on configuring the MPLS tunnel, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.

7. Repeat Steps 1 through 6 using unique values to configure the S-VLAN tunnel and MPLS tunnel interfaces on the remote E-series router. For example:

   host2(config)#**interface fastEthernet 3/1**
   host2(config-if)#**encapsulation vlan**
   host2(config-if)#**interface fastEthernet 3/1.1**
   host2(config-if)#**svlan id 83 any**
   host2(config-if)#**svlan ethertype 88a8**
   host2(config-if)#**route interface tunnel mpls:tunnel2 45**

encapsulation vlan

- Use to configure VLAN as the encapsulation method for the interface.
- Use the **no** version to disable VLAN on an interface.

interface fastEthernet

- Use to select a Fast Ethernet interface on a line module.
- Example

  host1(config)#**interface fastEthernet 3/1**

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

route interface

- Use to route layer 2 traffic on a specific tunnel interface.
- Use the **no** version to negate this command.

NOTE: For details on the use of this command, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN tunnel interface.
- Choose one of the following Ethertype values:
  - 8100—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - 88a8—Specifies Ethertype value 0x88a8, as defined in draft IEEE Standard 802.1ad
  - 9100—Specifies Ethertype value 0x9100, which is the default

- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-if)#**svlan ethertype 8100**

- Use the **no** version to restore the default value, 9100.

svlan id

- Use to create an S-VLAN tunnel interface for configuring Ethernet layer 2 services over MPLS.
- Assign an S-VLAN ID value in the range 0–4095 that is unique within the Ethernet interface.
- Use the **any** keyword to tunnel traffic from VLANs configured with the specified S-VLAN ID and any VLAN ID to the same destination across an MPLS network.
- Issue the **svlan id** command with the **any** keyword before you configure the upper binding, which must be an MPLS tunnel interface. Attempting to configure any other interface type over the S-VLAN tunnel causes an error.
- Example

  host1(config-if)#**svlan id 1000 any**

- There is no **no** version.

S-VLAN Oversubscription

When you configure S-VLAN subinterfaces over Ethernet interfaces to support dynamic PPPoE subinterfaces, you can take advantage of S-VLAN oversubscription.

The following module combinations support S-VLAN oversubscription:

- GE/FE line module and all of its associated I/O modules
- GE-2 line module and the GE-2 SFP I/O module
- GE-HDE line module and its associated I/O modules
- OC3/STM1 GE/FE line module and the OC3-2 GE APS I/O module
- ES2 4G LM and its associated Gigabit Ethernet and 10-Gigabit Ethernet IOAs
- ES2 10G LM and its associated Gigabit Ethernet and 10-Gigabit Ethernet IOAs

The maximum number of S-VLANs that you can create per I/O module with PPPoE major interfaces stacked over them is greater than the maximum number of dynamic PPPoE subinterfaces. The maximum number of PPP interfaces supported per line module is directly proportional to the maximum number of PPPoE subinterfaces.

As a result, you can oversubscribe S-VLANs by configuring up to the maximum number of S-VLANs supported on these I/O modules, knowing that no more than the maximum number of supported PPP sessions can be connected to the router at any one time.

For configuration instructions, see *Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns* in *JUNOSe Link Layer Configuration Guide, Chapter 15, Configuring Dynamic Interfaces*.

For specific information about the maximum number of S-VLANs supported per I/O module and the maximum number of PPP interfaces and PPPoE subinterfaces supported per line module, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: The E120 and E320 routers can support up to two IOAs per line module. This maximum number of S-VLANs per line module does not change if one or two IOAs are installed.

Monitoring VLAN and S-VLAN Subinterfaces

This section explains how to display bit rate and packet rate statistics for VLAN subinterfaces and use the **show** commands to display the physical characteristics and the configured settings for VLAN and S-VLAN subinterfaces.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Displaying Interface Rate Statistics for VLAN Subinterfaces

You can use the **monitor vlan interface** command to display bit rate and packet rate statistics over a specified time interval for one or more VLAN subinterfaces configured on the router.

To display interface rate statistics for VLAN subinterfaces:

1. Log in to the router by using a local console session or a virtual terminal (vty) session (such as a Telnet session).

   While you are using the **monitor vlan interface** command, you must keep the console or terminal session open and you cannot issue any other commands at the session during this time.

   For information about logging in to the router, see *Accessing the CLI* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

2. Access User Exec mode or Privileged Exec mode.

   For information, see *Accessing Command Modes* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

3. Specify the interface identifier for each VLAN subinterface that you want to monitor.

   host1#**monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1 display-time-of-day**

   For information about specifying interface identifiers for VLAN subinterfaces configured over Ethernet interfaces, see *VLAN Overview* on page 163. For information about specifying interface identifiers for VLAN subinterfaces configured over LAG bundles, see *Configuring a VLAN Subinterface for a LAG Bundle* on page 198.

   By default, the router uses a 5-second time interval between polls to calculate bit rates and packet rates for each specified VLAN subinterface. Optionally, you can use the **load-interval** keyword to specify a nondefault time interval in the range 5–30 seconds.

   You can also include the optional **display-time-of-day** keyword to show the time of day at which the router gathers statistics for each interval. Displaying the time of day enables you to monitor when a particular VLAN subinterface is underutilized or overutilized.

4. Review the command output.

```
host1#monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1
display-time-of-day
                       Seconds
                       between                                    Time
      Interface         polls    Input bps/pps   Output bps/pps   (UTC)
---------------------- -------- --------------- --------------- --------
FastEthernet 0/0.1            0           --/--           --/-- 10:50:07
FastEthernet 4/0.1            0           --/--           --/-- 10:50:07
FastEthernet 0/0.1            5      120240/100      120240/100 10:50:12
FastEthernet 4/0.1            5      120000/100      120000/100 10:50:12
FastEthernet 0/0.1            5      120240/100      120240/100 10:50:17
FastEthernet 4/0.1            5      120000/100      120000/100 10:50:17
```

   The router polls each VLAN subinterface at the specified load interval (the default 5-second interval in this example) to calculate and display bit rate and packet rate statistics. The first line of output for each interface always displays 0 (zero) for the number of seconds between polls, and dashes (*--/--*) in the Input bps/pps and Output bps/pps columns. These values indicate that the router initially takes a baseline for each interface against which to measure subsequent statistics. The router continues to display subsequent lines of output for each interface at the specified load interval until you press Ctrl + c to stop the command.

   For a description of each field in the **monitor vlan interface** command output, see **monitor vlan interface** on page 185.

5. When you are finished, press Ctrl + c to stop the **monitor vlan interface** command.

   host1#**^C**

monitor vlan interface

- Use to display bit rate and packet rate statistics over a specified time interval for one or more VLAN subinterfaces.
- You must use the **monitor vlan interface** command in a dedicated console or terminal session for the duration of the monitoring session.
- Specify the interface identifier for each VLAN subinterface that you want to monitor.
- To specify a nondefault time interval in the range 5–30 seconds at which the router calculates bit rate and packet rate statistics, use the optional **load-interval** keyword. The default time interval is 5 seconds.
- To display the time at which the router calculates bit rate and packet rate statistics for the current interval, use the optional **display-time-of-day** keyword.
- To stop the **monitor vlan interface** command, press Ctrl + c.
- Field descriptions
  - Interface—Interface identifier for the Ethernet or LAG interface on which the VLAN subinterface resides
  - Seconds between polls—Number of seconds at which the router calculates bit rate and packet rate statistics
  - Input bps/pps—Number of bits per second (bps) and packets per second (pps) received on this interface during the specified load interval
  - Output bps/pps—Number of bits per second (bps) and packets per second (pps) transmitted on this interface during the specified load interval
  - Time—Time of day, in hh:mm:ss format, at which the router calculates the bit rate and packet rate statistics for the current interval
- Example 1—Displays bit rate and packet rate statistics over the default (5-second) load interval for a single VLAN subinterface

```
host1#monitor vlan interface fastEthernet 0/0.1
                       Seconds
                       between
       Interface        polls    Input bps/pps   Output bps/pps
---------------------- -------- --------------- ---------------
FastEthernet 0/0.1             0           --/--           --/--
FastEthernet 0/0.1             5      120240/100      120240/100
FastEthernet 0/0.1             5      120000/100      120000/100
FastEthernet 0/0.1             5        92400/77        92400/77
FastEthernet 0/0.1             5        88800/74        88800/74
FastEthernet 0/0.1             5      120000/100      120000/100
host1#^C
```

- Example 2—Displays bit rate and packet rate statistics over a 10-second load interval for two VLAN subinterfaces, with the time of day that the statistics were calculated

```
host1#monitor vlan interface fastEthernet 0/0.1 fastEthernet 4/0.1
load-interval 10 display-time-of-day
                       Seconds
                       between                                      Time
       Interface        polls    Input bps/pps   Output bps/pps   (UTC)
----------------------- -------- --------------- --------------- --------
FastEthernet 0/0.1             0           --/--           --/-- 10:50:33
FastEthernet 4/0.1             0           --/--           --/-- 10:50:33
FastEthernet 0/0.1            10      120120/100      120120/100 10:50:43
FastEthernet 4/0.1            10      120000/100      120000/100 10:50:43
FastEthernet 0/0.1            10      120000/100      120000/100 10:50:53
FastEthernet 4/0.1            10      120000/100      120000/100 10:50:53
host1#^C
```

- There is no **no** version.

Using Ethernet show Commands

Use the **show** commands described in this section to display information about your Ethernet configuration and to monitor Ethernet interfaces.

show interfaces fastEthernet

- Use to display the status of Fast Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions when you display the status of a Fast Ethernet VLAN or S-VLAN subinterface
  - *Subinterface number*—Location of the subinterface that carries the VLAN or S-VLAN traffic
  - Administrative status—Operational state that you configured for this interface; up or down
  - VLAN ID—Domain number of the VLAN
  - SVLAN ID—Domain number of the stacked VLAN
  - Ethertype—Ethertype assignment for the S-VLAN subinterface, 0x8100, 0x88a8, or 0x9100; 0x9100 is the default
  - In—Analysis of inbound traffic on this interface
    - Bytes—Number of bytes received on the VLAN or S-VLAN subinterface
    - Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
    - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
    - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface

  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface
  - Packets—Number of packets sent on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets sent on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets sent on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all transmitted packets; note that some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets

- Example 1—Displays the status of a Fast Ethernet VLAN subinterface

```
host1:vr2#show interfaces fastEthernet 8/3.1
FastEthernet8/3.1 is Up, Administrative status is Up
  VLAN ID: 10, address 0090.5e00.0001

In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
Out: Bytes 4536220, Packets 70873
   Multicast 0, Broadcast 70258
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 1, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 1, ARP responses 0
   Errors 0, Discards 0
```

- Example 2—Displays the status of a Fast Ethernet S-VLAN subinterface

```
host1:vr2#show interfaces fastEthernet 0/0.1
FastEthernet0/0.1 is Up, Administrative status is Up
  SVLAN ID: 1, VLAN ID: 0, Ethertype 0x9100

  In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4536220, Packets 70873
   Multicast 0, Broadcast 70258
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

show interfaces gigabitEthernet
show interfaces tenGigabitEthernet

- Use to display the status of Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, VLAN subinterfaces, or S-VLAN subinterfaces.
- You can specify the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions when you display the status of a Gigabit Ethernet or 10-Gigabit Ethernet VLAN or S-VLAN subinterface
  - *Subinterface number*—Location of the subinterface that carries the VLAN or S-VLAN traffic
  - Administrative status—Operational state that you configured for this interface; up or down
  - VLAN ID—Domain number of the VLAN
  - SVLAN ID—Domain number of the stacked VLAN
  - Ethertype—Ethertype assignment for the S-VLAN subinterface, 0x8100, 0x88a8, or 0x9100; 0x9100 is the default
  - In—Analysis of inbound traffic on this interface
    - Bytes—Number of bytes received on the VLAN or S-VLAN subinterface
    - Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
    - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
    - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
    - Errors—Total number of errors in all received packets; some packets might contain more than one error
    - Discards—Total number of discarded incoming packets
  - Out—Analysis of outbound traffic on this interface
    - Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface
    - Packets—Number of packets sent on the VLAN or S-VLAN subinterface
    - Multicast—Number of multicast packets sent on the VLAN or S-VLAN subinterface
    - Broadcast—Number of broadcast packets sent on the VLAN or S-VLAN subinterface
    - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
    - Discards—Total number of discarded outgoing packets

- Example 1—Displays the status of a Gigabit Ethernet VLAN subinterface

```
host1:vr2#show interfaces gigabitEthernet 2/0.1
GigabitEthernet2/0.1 is Up, Administrative status is Up
  VLAN ID: 10, address 0090.5e00.0001

  In: Bytes 2357, Packets 23
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4872, Packets 57
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

- Example 2—Displays the status of a Gigabit Ethernet S-VLAN subinterface

```
host1:vr2#show interfaces gigabitEthernet 2/0.2
GigabitEthernet2/0.2 is Up, Administrative status is Up
  SVLAN ID: 10, VLAN ID: 100, Ethertype 0x9100

  In: Bytes 2357, Packets 23
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4872, Packets 57
   Multicast 0, Broadcast 57
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

show vlan subinterface

- Use to display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router.
- Use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router.
- Use the **mac-address** keyword to display information about the VLAN subinterfaces that were configured with unique MAC addresses.
- Use the **vlan** or **svlan** keywords to display information about specific S-VLAN IDs or VLAN IDs.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface
  - Status—Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
  - MTU—Maximum allowable size (in bytes) of the maximum transmission unit (MTU) for the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID value for the VLAN subinterface
  - Ethertype—S-VLAN Ethertype value, if configured

- Type—Type of VLAN subinterface
  - Static—VLAN or S-VLAN subinterface was configured statically
  - Dynamic—VLAN or S-VLAN subinterface was configured dynamically
- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received on the VLAN or S-VLAN subinterface
  - Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface
  - Packets—Number of packets sent on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- Total VLAN interfaces—Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; this is the only field that appears when you specify the **summary** keyword

- Example 1—Displays full status and configuration information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
ATM 3/0.1.2              Up         1522 ----     11       ----      Static
ATM 3/0.1.3              Up         1522 ----     12       ----      Static
ATM 3/1.1.1              Up         1522 ----     13       ----      Static
ATM 3/1.1.2              Up         1522 ----     14       ----      Static
ATM 3/2.1.1              Down       1526 4        255      0x9100    Static
FastEthernet 4/5.1       Up         1522 ----     1        ----      Dynamic
6 vlan subinterfaces found
```

- Example 2—Displays full status and configuration information for the specified VLAN subinterface

```
host1#show vlan subinterface fastEthernet 0/0.1
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 0/0.1       Up         1526       1        0    0x9100 Static

In: Bytes 39256, Packets 612
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 4538652, Packets 70911
   Multicast 0, Broadcast 70296
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

- Example 3—Displays only brief summary information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface summary
Total VLAN interfaces: 6 subinterfaces, 3 major interfaces
```

- Example 4—Displays full status and configuration information for all VLAN subinterfaces configured with a unique MAC address

```
host1#show vlan subinterface mac-address
        Interface          Svlan Id  Vlan Id    MAC Address
------------------------  --------  --------  --------------
FastEthernet 4/0.25            ----        25  0090.dfad.2abd
FastEthernet 4/0.10050            1        50  0090.adad.0abd
2 vlan subinterfaces found
```

- Example 5—Displays full status and configuration information for a VLAN subinterface on a LAG bundle

```
host1#show vlan subinterface lag boston.1
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
lag boston.1             Up         1522 ----     1        ----      Static
```

- Example 6—Displays full status and configuration information for the specified S-VLAN ID

```
host1#show vlan subinterface svlan 100 53
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 0/0.1        Up         1526 100      53       0x9100    Static
FastEthernet 4/6.1        Up         1526 100      53       0x9100    Dynamic
2 vlan subinterfaces found
```

Chapter 6

Configuring 802.3ad Link Aggregation and Link Redundancy

This chapter describes how to configure 802.3ad link aggregation and link redundancy on E-series routers.

This chapter contains the following sections:

802.3ad Link Aggregation for Ethernet Overview

IEEE 802.3ad link aggregation enables you to group Ethernet interfaces at the physical layer to form a single link layer interface, also known as a link aggregation group (LAG) or bundle. For more information, see IEEE Standard 802.3ad, Link Aggregation.

Some users require more bandwidth in their network than a single Fast Ethernet link can provide, but cannot afford the expense or do not need the bandwidth of a higher-speed Gigabit Ethernet link. Using IEEE 802.3ad link aggregation in this situation provides increased port density and bandwidth at lower cost. For example, if you need 450 Mbps of bandwidth to transmit data and have only a 100-Mbps Fast Ethernet link, creating a LAG bundle containing five 100-Mbps Fast Ethernet links is more cost effective than purchasing a single Gigabit Ethernet link.

For information about the modules that support link aggregation, see *ERX Module Guide, Appendix A, Module Protocol Support* and *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

LACP

The Link Aggregation Control Protocol (LACP) is a mechanism for exchanging port and system information to create and maintain LAG bundles. The LAG bundle distributes MAC clients across the link layer interface and collects traffic from the links to present to the MAC clients of the LAG bundle.

To create the links in the LAG bundles, you can add one or more Ethernet physical interfaces to it. The LACP detects Ethernet interfaces as links if they are configured on the same line module and have the same physical layer characteristics. The LACP also assigns to the LAG bundle the same MAC address of the Ethernet link with the highest port priority, which is the lowest value.

The LACP also controls the exchange of LACP protocol data units (PDUs) between the Ethernet links in the LAG bundle. The PDUs contain information about each link and enable the LAG bundle to maintain them.

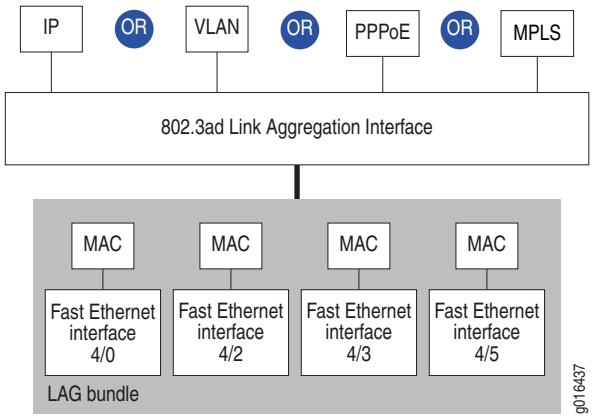
By default, Ethernet links do not exchange PDUs, which contain information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or passively transmit them, sending out LACP PDUs only when it receives them from another link. The transmitting link is known as the *Actor* and the receiving link is known as the *Partner*.

Higher-Level Protocols

After you configure the LAG bundle, you can route IP traffic over it, create a VLAN over it, route PPPoE traffic over it, or route MPLS traffic over it.

Figure 20 displays the interface stack for 802.3ad link aggregation.

Figure 20: Interface Stack for 802.3ad Link Aggregation



For information about configuring higher-level protocols over VLANs, see *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces*.

NOTE: On the ES2 10G LM and ES2-S1 GE-8 IOA combination, you can only configure IP or VLAN over a LAG bundle.

Load Balancing and QoS

You can configure load balancing across 802.3ad links to provide quality of service (QoS). To ensure that QoS is symmetrically applied to all the links, the router periodically rebalances the traffic on the LAG. When you attach a QoS profile to the LAG, the load balancing properties that are configured are applied to the LAG, and determines how traffic is distributed.

For example, if VLANs are configured, IP queues are provisioned over the VLANs. In this case, the default behavior is per-VLAN load balancing.

For more information, see *JUNOSe Quality of Service Configuration Guide, Chapter 20, Configuring QoS for Gigabit Ethernet Interfaces and VLAN Subinterfaces*.

Ethernet Link Aggregation and MPLS

CE-side load balancing in a Martini layer 2 transport environment enables an E-series router to interoperate with an 802.3ad switch in a topology designed for Ethernet link aggregation. See *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS* for more information.

802.3ad Link Aggregation Platform Considerations

You can configure 802.3ad link aggregation on the following E-series routers:

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

Module Requirements

For information about the modules that support 802.3ad link aggregation on ERX-14xx models, ERX-7xx 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 802.3ad link aggregation.

For information about the modules that support 802.3ad link aggregation 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 802.3ad link aggregation.

Interface Specifiers

The configuration task examples in this chapter use the format for ERX-7xx models, ERX-14xx models, and the ERX-310 router to specify 802.3ad link aggregation.

For example, the following command specifies a Gigabit Ethernet interface on port 0 of an I/O module in slot 4.

```
host1(config)#interface gigabitEthernet 4/0
```

When you configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface on E120 or E320 routers, you must include the adapter identifier as part of the interface specifier. For example, the following command specifies a Gigabit Ethernet interface on port 0 of the IOA installed in the upper adapter bay of slot 3.

```
host1(config)#interface gigabitEthernet 3/0/0
```

For more information about interface types and specifiers on E-series models, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

802.3ad Link Aggregation References

For more information about 802.3ad link aggregation implementations, consult the following resources:

- IEEE 802.1w (Rapid Reconfiguration of Spanning Tree)
- IEEE 802.3ad (Link Aggregation)

Configuring 802.3ad Link Aggregation

To configure link aggregation on Ethernet interfaces, you must configure the Ethernet interface, create the LAG bundle, and add the Ethernet interface as a member link in the LAG bundle. Optionally, you can then configure IP, a VLAN subinterface, a PPPoE subinterface, or MPLS for the LAG bundle.

For more information about specifying LAG interfaces and subinterfaces on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Tasks to configure 802.3ad link aggregation interfaces are:

- Configuring an Ethernet Physical Interface on page 197
- Configuring a LAG Bundle on page 198
- Configuring IP for a LAG Bundle on page 198
- Configuring a VLAN Subinterface for a LAG Bundle on page 198
- Configuring a PPPoE Subinterface for a LAG Bundle on page 199
- Configuring MPLS for a LAG Bundle on page 199

Configuring an Ethernet Physical Interface

To configure a member link, perform the following steps:

1. Specify a Fast Ethernet or Gigabit Ethernet interface for which you want to create a member link.

   host1(config)#**interface gigabitEthernet 2/0**

2. Configure LACP in passive or active mode.

   host1(config-if)#**lacp active**

3. Specify the speed and the duplex mode for the Ethernet interface.

   host1(config-if)#**speed 100**
   host1(config-if)#**duplex full**

4. Specify the MTU.

   host1(config-if)#**mtu 9000**

5. To configure additional member links, repeat steps 1 to 4.

NOTE: All of the member links that you configure must be on the same line module and have the same physical layer characteristics, such as speed, duplex mode, and MTU.

Configuring a LAG Bundle

To configure a LAG bundle and add member links, perform the following steps:

1. Create the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Add a member link to the LAG bundle.

   host1(config-if)#**member–interface gigabitEthernet 2/0**

Configuring IP for a LAG Bundle

To configure IP for a LAG bundle, perform the following steps:

1. Specify the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Assign an IP address and mask.

   host1(config-if)#**ip address 192.5.127.8 255.255.255.0**

Configuring a VLAN Subinterface for a LAG Bundle

To configure a VLAN subinterface for the LAG bundle, perform the following steps:

1. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

2. Specify the VLAN subinterface for the LAG bundle by adding a unique subinterface number to the LAG interface identification command.

   host1(config)# **interface lag bundleBoston.1**

3. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 203**

4. Assign an IP address and mask.

   host1(config-if)#**ip address 192.168.1.1 255.255.0.0**

Configuring a PPPoE Subinterface for a LAG Bundle

To configure a PPPoE subinterface for the LAG bundle, perform the following steps:

1. Specify PPPoE as the encapsulation method.

   host1(config-if)#**encapsulation pppoe**

2. Specify the PPPoE subinterface for the LAG bundle in either of the following ways:

   - Use the **interface lag** command to add a unique subinterface number to the LAG bundle name.

     host1(config)#**interface lag bundleBoston.2**

   - Use the **pppoe subinterface lag** command to add a unique subinterface number to the LAG bundle name.

     host1(config)#**pppoe subinterface lag bundleBoston.2**

3. Specify PPP as the encapsulation method on the PPPoE subinterface.

   host1(config-if)#**encapsulation ppp**

4. Assign an IP address and mask.

   host1(config-if)#**ip address 192.168.1.2 255.255.0.0**

You can also configure a PPPoE subinterface over a VLAN subinterface over a LAG bundle. For an example of this configuration, see *Example: Configuring a PPPoE Subinterface over a VLAN for a LAG Bundle* on page 203.

Configuring MPLS for a LAG Bundle

To configure MPLS for a LAG bundle, perform the following steps:

1. Specify the LAG bundle.

   host1(config)#**interface lag bundleBoston**

2. Create an MPLS interface.

   host1(config-if)#**mpls**

interface lag

- Use to create an IEEE 802.3ad LAG interface, also known as a LAG bundle, or a subinterface for the LAG bundle.
- Examples

  host1(config)#**interface lag boston**
  host1(config)#**interface lag boston.2**
  host1(config)#**interface lag boston.2.1**

- Use the **no** version to delete the LAG bundle.

lacp

- Use to configure whether an Ethernet link in a LAG bundle participates actively or passively in the LACP.
- Use the **active** keyword to indicate that the Ethernet link participates in the protocol regardless of whether its Partner member link is set to active or passive LACP PDU participation.
- Use the **passive** keyword to indicate that the Ethernet link to transmit LACP PDUs only when it receives LACP PDUs from its Partner member link.
- By default, Ethernet links in a LAG bundle do not send LACP PDUs.
- Example

  host1(config-if)#**lacp active**

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

lacp port-priority

- Use to set the priority for an Ethernet link in a LAG bundle.
- The member with the lowest value has the highest priority, and is selected to join the LAG bundle first.
- Valid values are in the range 0–65535.
- Example

  host1(config-if)#**lacp port-priority 100**

- Use the **no** version to restore the default value of 32768.

member-interface

- Use to add a Fast Ethernet interface or Gigabit Ethernet interface, also known as a bundle member, to a LAG bundle.
- Example

  host1(config-if)#**member-interface fastEthernet 4/0**

- Use the **no** version to remove the specified Ethernet link from the bundle.

mpls

- Use to enable, disable, or delete MPLS on an interface. MPLS is disabled by default.
- Example

  host1(config)#**mpls**

- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.

mtu

- Use to specify the MTU for a LAG bundle.
- Specify a value in the range 64–9188 bytes. The range for FE-8 I/O modules is 64–9042 bytes.
- This command does not work for the Fast Ethernet port on the SRP module.
- Example

  host1(config-if)#**mtu 9000**

- Use the **no** version to specify the default, 1518.

pppoe subinterface lag

- Use to create a PPPoE subinterface on a LAG bundle.
- Example

  host1(config-if)#**pppoe subinterface lag boston.1**

- Use the **no** version to remove the PPPoE subinterface from the LAG bundle.

virtual-router

- From Global Configuration mode, use this command to create a virtual router or access the context of a previously created virtual router or a VRF.
- Example

  host1(config)#**virtual-router boston**

- Use the **no** version of the command only to delete the VR and return the router to the default VR.

Example: Configuring an IP Interface for a LAG Bundle

The following example displays configuration of LACP for two Fast Ethernet interfaces in slot 0. The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both interfaces.

```
host1(config)#interface fastEthernet 0/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
host1(config-if)#interface fastEthernet 0/5
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
```

The following commands create a virtual router, add the Ethernet physical interfaces to a LAG bundle named bundleBoston, and assign an IP address and mask to the bundle.

```
host1(config)#virtual-router boston
host1:boston(config)#interface lag boston
host1:boston(config-if)#member-interface fastEthernet 0/0
host1:boston(config-if)#member-interface fastEthernet 0/5
host1:boston(config-if)#ip address 1.1.1.1 255.255.255.0
```

Example: Configuring a PPPoE Subinterface for a LAG Bundle

The following example displays LACP configuration for two Fast Ethernet interfaces in slot 4. The interfaces are enabled for passive LACP. The speed and duplex characteristics are the same for both interfaces.

```
host1(config)#interface fastEthernet 4/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp passive
host1(config-if)#interface fastEthernet 4/3
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp passive
```

The following commands add the Ethernet physical interfaces to a LAG bundle named chicago.

```
host1(config)#interface lag chicago
host1(config-if)#member-interface fastEthernet 4/0
host1(config-if)#member-interface fastEthernet 4/3
```

The following commands configure a PPPoE subinterface for the LAG bundle named chicago. In the LAG interface identification command (**interface lag chicago.1**), the number 1 represents the subinterface number for the PPPoE subinterface.

```
host1(config-if)#encapsulation pppoe
host1(config)#interface lag chicago.1
host1(config-if)#encapsulation ppp
host1(config-if)#ip address 10.10.1.1 255.255.0.0
```

As an alternative to using the command **interface lag chicago.1** to configure the PPPoE subinterface in this example, you can also use the command **pppoe subinterface lag chicago.1** to achieve the same result. For more information, see **pppoe subinterface lag** on page 201.

Example: Configuring a PPPoE Subinterface over a VLAN for a LAG Bundle

The following example displays LACP configuration for two Fast Ethernet interfaces in slot 3. The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both interfaces.

```
host1(config)#interface fastEthernet 3/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
host1(config-if)#interface fastEthernet 3/1
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
```

The following commands add the Ethernet physical interfaces to a LAG bundle named sunnyvale.

```
host1(config)#interface lag sunnyvale
host1(config-if)#member-interface fastEthernet 3/0
host1(config-if)#member-interface fastEthernet 3/1
```

The following commands configure a VLAN subinterface for the LAG bundle named sunnyvale. In the LAG interface identification command (**interface lag sunnyvale.1**), the number 1 represents the subinterface number for the VLAN subinterface.

```
host1(config-if)#encapsulation vlan
host1(config)#interface lag sunnyvale.1
host1(config-if)#vlan id 100
```

The following commands configure a PPPoE subinterface over the VLAN subinterface for the LAG bundle named sunnyvale. In the LAG interface identification command (**interface lag sunnyvale.1.2**), the number 2 represents the subinterface number for the PPPoE subinterface.

```
host1(config-if)#encapsulation pppoe
host1(config)#interface lag sunnyvale.1.2
host1(config-if)#encapsulation ppp
host1(config-if)#ip address 10.10.2.2 255.255.0.0
```

As an alternative to using the command **interface lag sunnyvale.1.2** to configure the PPPoE subinterface in this example, you can also use the command **pppoe subinterface lag sunnyvale.1.2** to achieve the same result. For more information, see **pppoe subinterface lag** on page 201.

Example: Configuring MPLS for a LAG Bundle

The following example displays configuration of LACP for two Fast Ethernet interfaces in slot 5. The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both interfaces.

```
host1(config)#interface fastEthernet 5/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
host1(config-if)#interface fastEthernet 5/1
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
```

The following commands create a virtual router, add the Ethernet physical interfaces to a LAG bundle named kanata, assign an IP address, and configure MPLS.

```
host1(config)#virtual router kanata
host1:kanata(config)#interface lag kanata
host1:kanata(config-if)#member-interface fastEthernet 0/0
host1:kanata(config-if)#member-interface fastEthernet 0/5
host1:kanata(config-if)#ip address 1.1.1.1 255.255.255.0
host1(config-if)#mpls
```

Example: Configuring MPLS over a VLAN for a LAG Bundle

The following example displays configuration of LACP for two Fast Ethernet interfaces in slot 5. The interfaces are enabled for active LACP. The speed and duplex characteristics are the same for both interfaces.

```
host1(config)#interface fastEthernet 5/0
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
host1(config-if)#interface fastEthernet 5/1
host1(config-if)#speed 100
host1(config-if)#duplex full
host1(config-if)#lacp active
```

The following commands add the Ethernet physical interfaces to a LAG bundle named kanata.

```
host1(config)#virtual router kanata
host1:kanata(config)#interface lag kanata
host1:kanata(config-if)#member-interface fastEthernet 5/0
host1:kanata(config-if)#member-interface fastEthernet 5/1
```

The following commands configure a VLAN subinterface for the LAG bundle named kanata. In the LAG interface identification command (**interface lag kanata.1**), the number 1 represents the subinterface number for the VLAN subinterface.

```
host1:kanata(config-if)#encapsulation vlan
host1:kanata(config)#interface lag kanata.1
host1:kanata(config-if)#vlan id 100
```

The following command creates an MPLS interface.

```
host1:kanata(config)#mpls
```

Ethernet Link Redundancy Overview

You can use 802.3ad Link Aggregation (LAG) to configure Ethernet link redundancy for Fast Ethernet and Gigabit Ethernet interfaces. Ethernet link redundancy enables you to protect against physical link failure and account for network topology changes that redirect network traffic to redundant ports.

The following configurations are available:

- LAG to LAG—Provides redundancy capabilities for two or more ports that are assigned to a LAG. One member link is configured as the backup interface for all other ports in the LAG bundle (1:N). Traffic is not forwarded over the backup member interface; it is disabled until it takes over for an active member interface.

- LAG to non-LAG—Provides redundancy capabilities when redundant ports are connected to a bridged network that has Rapid Spanning Tree Protocol (RSTP) controlling the topology. This configuration supports only two links in the LAG.

For information about the modules that support link aggregation, see *ERX Module Guide, Appendix A, Module Protocol Support* and *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

Ethernet Link Redundancy Configuration Models

The link connections determine the configuration model for link redundancy. The following connection types are available:

- Single-homed—Connections are between the local Ethernet interface and a single remote device. When the peer is also configured with LAG, LACP can be used to control link access.

- Dual-homed—Connections are between two separate, uncoordinated remote devices. The remote interfaces can be on the same module or on separate hardware. If LAG is not configured on the peers, LACP cannot be used to select ports; other protocols such as RSTP can be used.

The type of hardware used for connections further characterizes the single-homed and dual-homed configuration models. The following hardware types are available:

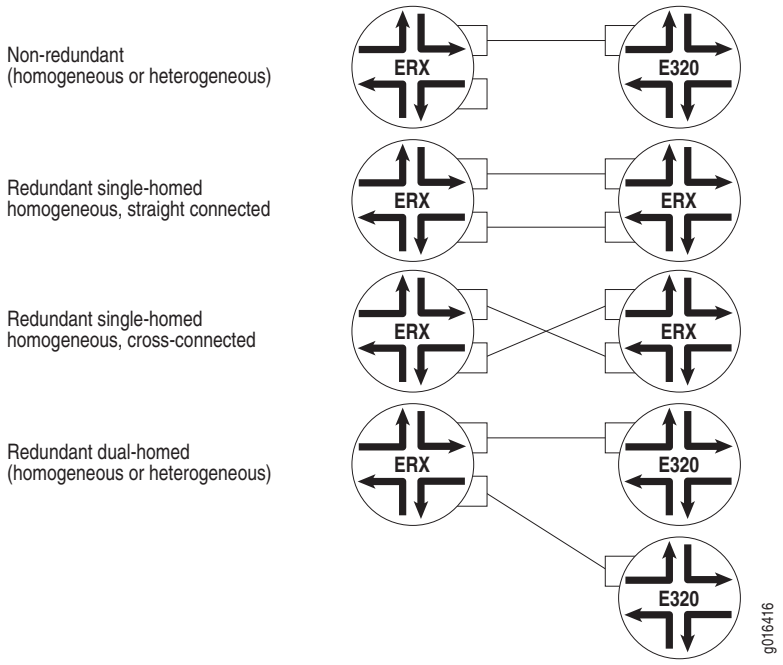
- Homogeneous—Remote interface is on another Fast Ethernet or Gigabit Ethernet port in a back-to-back router configuration of identical hardware and JUNOSe software versions. Both interfaces support the same redundant cabling and algorithm. The interfaces can be cabled on the same ports (port 0–port 0, port 1–port 1) or cross-cabled (port 0–port 1, port 1–port 0).

- Heterogeneous—Remote interface is on a different type of hardware that might or might not support redundant cabling, or on the same type of equipment with different software versions. For example, a heterogeneous configuration can include an ES2-S1 GE-4 IOA and an ES2-S1 GE-8 IOA on the E320 router, or an E-series router operating JUNOSe software connected to another vendor's router and software.

NOTE: You cannot configure link redundancy across different types of line modules in a router. You also cannot configure link redundancy across two GE-4 IOAs on the E120 router or the E320 router.

Figure 21 illustrates the configuration models for Ethernet link redundancy.

Figure 21: Ethernet Link Redundancy Configuration Models



Ethernet Link Redundancy Configuration Diagrams

The diagrams in this section illustrate examples of Ethernet link redundancy configurations. The diagrams display adjacent ports bundled in a LAG.

GE-2 Line Module Configurations

These diagrams compare physical port redundancy and link redundancy on a GE-2 line module.

Figure 22 displays a GE-2 line module with physical port redundancy on both ports.

Figure 22: GE-2 Line Module Using Physical Port Redundancy

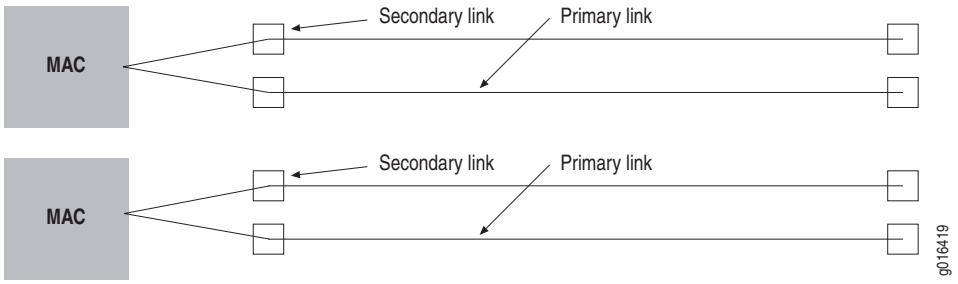
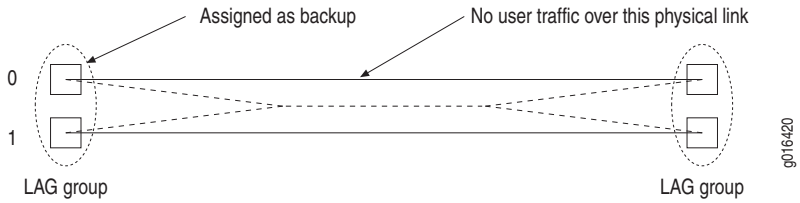


Figure 23 displays a single-homed configuration with port 0 backing up port 1 on a GE-2 line module.

Figure 23: Single-Homed GE-2 Line Module Configuration



FE-8 Line Module Configurations

Figure 24 displays an FE-8 line module with a link failure in a 1:N single-homed configuration.

Figure 24: Single-Homed FE-8 Line Module Configuration (1:N)

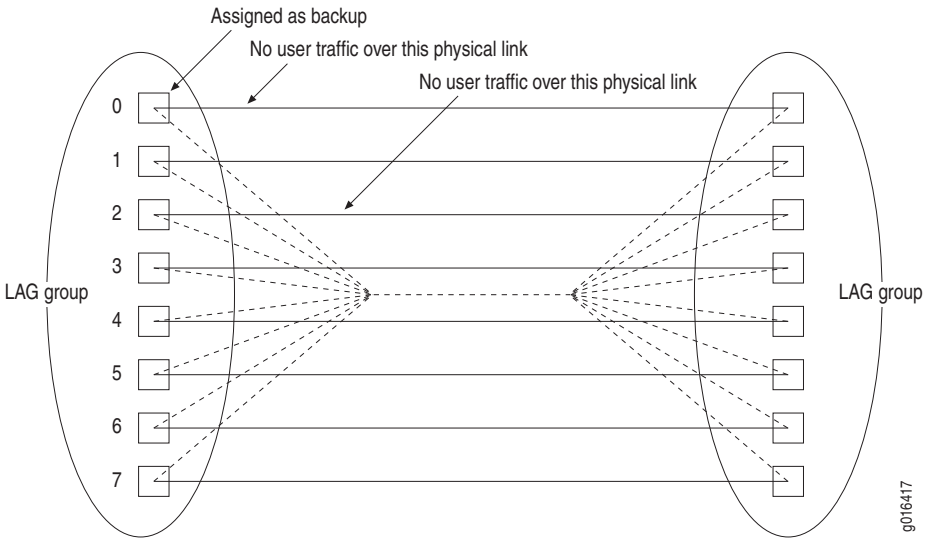
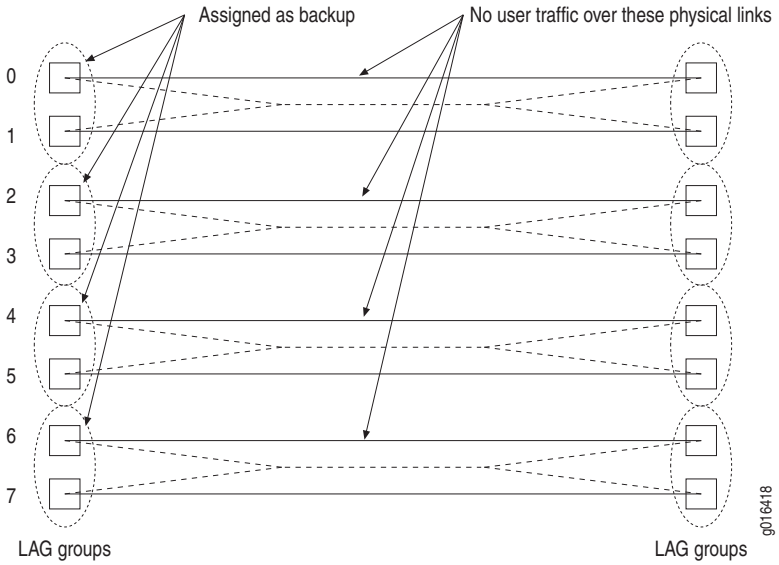


Figure 25 displays an FE-8 line module with four redundant Ethernet links in a 1:1 configuration.

Figure 25: FE-8 Line Module with 4 Redundant Ethernet Links (1:1)



E120 and E320 Router Configurations

Figure 26 and Figure 27 display link redundancy configurations on the E120 and E320 routers.

Figure 26 displays a single-homed 1:4 configuration on an E120 router.

Figure 26: Single-Homed GE-4 IOA Configuration (1:4)

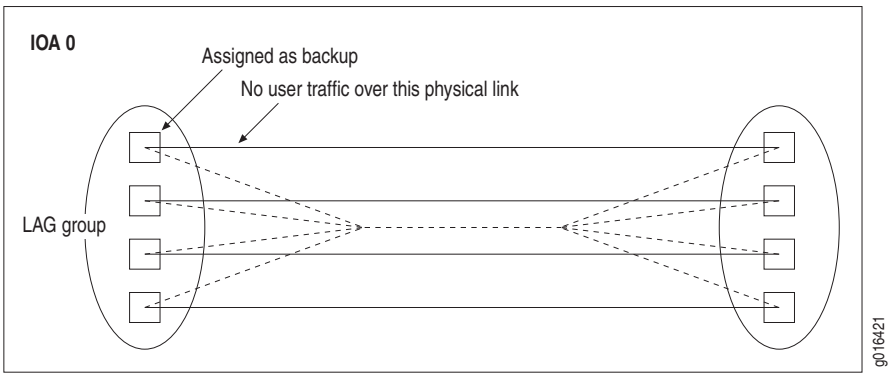
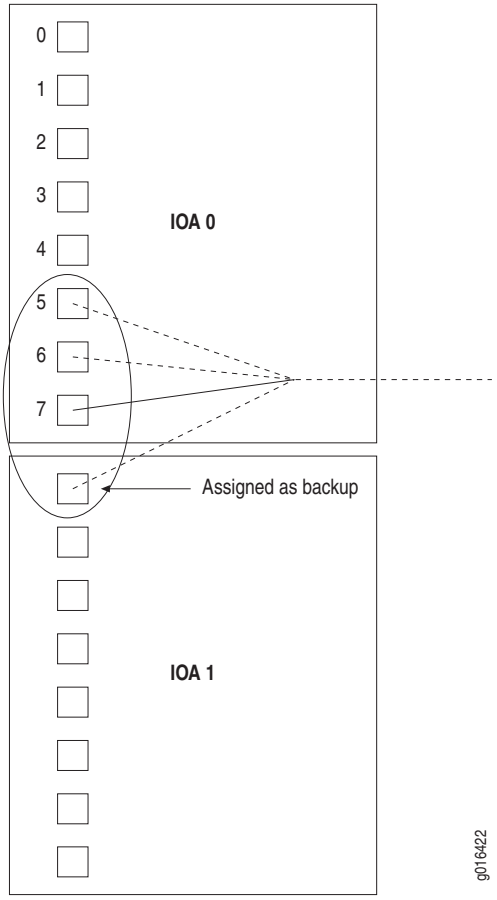


Figure 27 displays an E320 router with 1:N configuration across IOAs.

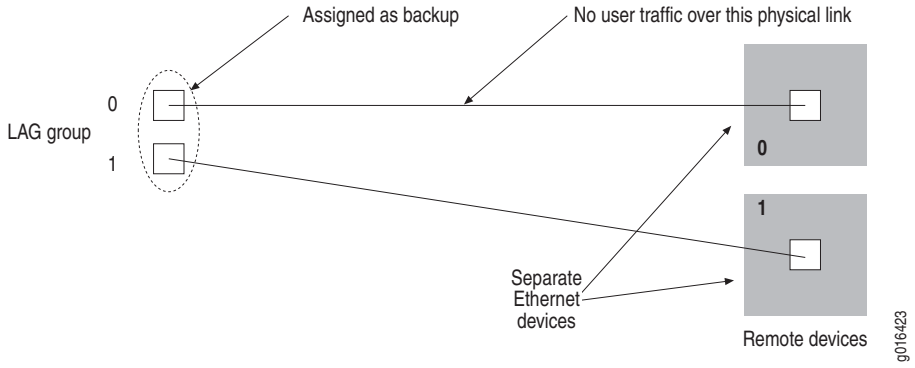
Figure 27: GE-8 IOA Configuration Across IOAs (1:N)



Dual-Homed Configurations with LAG Disabled

Figure 28 displays how you can configure Ethernet link redundancy with LACP disabled locally using a dual-homed configuration. LACP is disabled because there is no LAG at the peer.

Figure 28: Dual-Homed Configuration (1:1)



Ethernet Link Redundancy Behavior

When you create a LAG bundle, you can configure LACP with the Disabled, Passive, or Active states. For more information about these states, see *LACP* on page 194.

The following sections describe link redundancy behavior when the:

- Configuration and status of LACP changes during link failure and acquisition.
- Configuration of the endpoints of the member links is different.
- Configuration is LAG to non-LAG in an RSTP network.

Link Failure and Acquisition

Link failure on the local system occurs when the active link is no longer active. Failures can be characterized as physical link failure or virtual link failure.

Each type of link failure has different requirements for detection, failover, and link acquisition. In all cases, you configure the link to fail over when it fails by issuing the **redundant-port** command. Optionally, you can force the failover automatically by issuing the **redundant-port force-failover** command.

Protecting Against Physical Link Failure

Physical link failures can occur when a cable is cut.

To protect against physical link failure, issue the **transmitter** keyword with the **redundant-port** command to enable or disable the local redundant link. When the redundant link needs to be down, the link behavior in failure detection and failover follows a similar port redundancy scheme available with line modules such as the GE-2 line module. Disabling the transmitter also enables the remote end of the redundant link to be in the operational Down state, which might be a requirement for third-party equipment when supporting redundancy over LAG.

Enabling the transmitter provides for a quick LAG failover in the event one of the non-redundant links in the LAG fail. This is particularly true when LACP has been enabled on the LAG, because it can take several seconds for LACP to converge on a link. When the transmitter on the remote end is enabled on the redundant link before it fails over, the local system considers the redundant link to be viable and enables the transmitter if it is disabled. If the remote end is disabled, the local end must enable the transmitter and wait for the remote end to enable.

Protecting Against Virtual Link Failure

A virtual link failure can occur when the active link is no longer used by the network because of topology changes caused by physical failure in the network. Topology changes can occur when, for example, a link is blocked because of network protocols such as RSTP blocking the port leading to selection of the redundant port connected to the receiver.

To protect against virtual link failure in conjunction with network protocols, use the **packet-sampling** keyword with the **redundant-port** command to detect link the viability. For example, when there is a network protocol decision that changes the topology and blocks a link to compensate for failures in the network, the system monitors the traffic to detect the change in network topology and fails over to the redundant port if necessary. It also determines whether the failover is successful. For more information, see *Member Link with Non-LAG Partner* on page 212.

Reverting After a Failover

When you specify the **auto-revert** keyword with the **redundant-port** command, the redundant link reverts back to redundant mode when the failed link becomes active again.

The system uses the following process when you issue the **auto-revert on** and **auto-revert off** keywords:

auto-revert on

1. An active link fails and a redundant link becomes active.
2. The original active link becomes active.
3. The original redundant link fails over to the original active link.
4. The redundant link can fail over to any other active link again.

auto-revert off

1. An active link fails and a redundant link becomes active.
2. The original active link becomes active.
3. The original redundant link remains the active link.
4. You can force the link to fail over by issuing the **redundant-port force-failover** command.

LACP Configuration and Member Link Behavior

By default, when a redundant member link is configured, the system disables LACP and the transmitter on that link.

When a member link is administratively down, the link state is operationally down at the local and remote ends, which means it does not transmit or receive PDUs.

The active link does not fail over when:

- An active link goes down and you set the redundant member link to administratively down.
- An active link is set to administratively down.

LACP configurations affect member link behavior based on the local or remote endpoint. For a remote end to include a member link in link aggregation, the two member links that are connected must have LACP configured.

Table 11 lists the acceptable configurations that enable redundant behavior for LACP modes at local and remote endpoints.

Table 11: Behavior of Member Links Using Local and Remote LACP Modes

		Remote LACP Mode		
		Disabled	**Passive**	**Active**
Local LACP Mode	Disabled	✔	✔	–
	Passive	✔	✔	✔
	Active	–	✔	✔

Member Link with Non-LAG Partner

When a member link has a non-LAG partner, there are two separate links in a 1:1 configuration. To successfully configure this, you must disable LACP.

When a failover occurs and LACP is active, the partner might receive a new LAG ID and the LACP PDUs receive a new MAC address; therefore, the member links are not aggregated or the bundle is disabled, terminating the sessions above it.

The partner that is connected to the redundant link must not be forwarding network traffic; that is, it is either blocked through a protocol such as RSTP, or MAC address learning has selected the active port. The redundant link must not transmit over the redundant link to that MAC. The behavior of the redundant link depends on the failure detection method that is controlled by the network protocol that is blocking the port.

Ethernet Link Redundancy and RSTP

In a LAG to non-LAG configuration, you can configure redundancy capabilities when redundant ports are connected to a bridged network that has RSTP controlling the topology.

On external devices, we recommend that you configure RSTP-enabled bridged ports that are connected to the LAG interfaces as edge ports to enable the ports to transition quickly to forwarding state upon reconfiguration, and to avoid the listening and learning states required by the spanning tree protocol. The edge port designation instructs the local bridge that bridge loops do not exist through the interface, enabling it to skip the listening and learning states.

Figure 29: Dual-Homed Heterogeneous Configuration in an RSTP Network

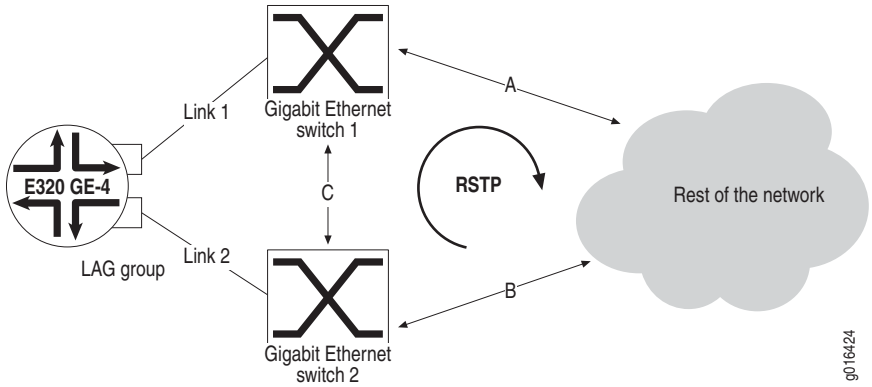


Figure 29 displays a network with RSTP enabled on Gigabit Ethernet switches 1 and 2. The local port receives bridge PDUs (BPDU), Ethernet broadcasts, and flooded unicast packets. If Link 1 is initially active and Link 2 is the backup, initial traffic destined for the LAG can be Ethernet broadcasts, PPPoE PDUs, or flooded Ethernet unicasts. The responses are only sent on the active link; in this case, Link 1.

The Ethernet network topology that is managed by RSTP learns that the MAC for the LAG group is through Link 1. Broadcasts and flooded packets are still sent on Link 2. If Link 1 is no longer viable, but has not suffered a physical failure, then that address ages out of the bridge databases and any packets directed to the LAG are flooded. The LAG detects traffic on Link 2 after the minimum delay time and then fails over.

Acquiring Initial Links

In an RSTP network, the system uses the following process for acquiring new links:

1. Based on the configuration, the system selects a link as active and the other as redundant.

2. The spanning tree converges on a topology.

3. When convergence occurs and the status of the spanning tree ports change to forwarding, network traffic appears on the links.

4. The local port detects the traffic and confirms the active member as active and the other as the redundant port. Because the initial traffic is broadcast or flooded, both ports receive the packets. However, because of the timing difference, the selected active port remains active.

Detecting Failures

In an RSTP network, the system uses the following process for detecting when the link has switched over due to topology changes:

1. BPDUs are ignored on the redundant port and system time is not retrieved. Because MAC learning forces non-flooded unicast packets to the active link, traffic to the redundant link does not receive non-flooded packets. The most recent system time is always retrieved when a network packet is received.

2. When the network cannot reach the active link because of topology changes, traffic appears on the redundant link. The redundant port detects the traffic and captures the latest timestamp. When the difference between the timestamp of the first non-bridged PDU and the time the last packet that was received on the active port is sufficiently large to account for the minimum spanning tree convergence time and latency for flooded and broadcast packets, then the port fails over.

Failing Over

In an RSTP network, the system uses the following process to fail over:

1. When the link has failed over, the system monitors the previously active port.

2. When a network packet is received on the redundant port, the system retrieves the timestamp. If the difference in timestamps between that one and the most recent on the current active port is more than the configured failover delay time, then the link fails over. If the difference is less than the delay time, the system ignores it but counts the event. If many of these transitions occur in a time period, then the system administratively brings the ports down. If no network traffic is received on either port, then failover does not occur.

Configuring Ethernet Link Redundancy

To configure Ethernet link redundancy:

1. Specify the Fast Ethernet or Gigabit Ethernet interface on which to configure a redundant link.

   ```
   host1(config)#interface gigabitEthernet 1/1
   ```

2. For LAG to non-LAG configurations only, specify that LACP is disabled on the port.

   ```
   host1(config-if)#no lacp
   ```

3. Configure a backup interface and disable LACP on it.

   ```
   host1(config)#interface gigabitEthernet 1/0
   host1(config-if)#no lacp
   ```

4. Configure a LAG interface and assign a member link to the backup interface.

   host1(config)#**interface lag myBundle**
   host1(config-if)#**member-interface gigabitEthernet 1/0**
   host1(config-if)#**member-interface gigabitEthernet 1/1**

5. Do one of the following:

   - Configure link redundancy on the port you specified in step 1.

     host1(config-if)#**redundant-port gigabitEthernet 1/1**

   - Force the port you specified in step 1 to fail over.

     host1(config-if)#**redundant-port gigabitEthernet 1/1 force-failover**

redundant-port

- Use to specify a member link in a LAG bundle as redundant.
- Use the **failover timeout** keyword to configure the amount of time between the current link event leading to failover or reversion and the previous link failover or reversion.
- Use the **packet-sampling** keyword to configure redundancy on a LAG to non-LAG application where packet sampling is used for failover detection. Use the optional **delay** keyword to control the minimum time difference to force packets on the active and redundant port to fail over.
- Use the **transmitter** keyword to enable or disable the transmitter when in redundant mode.
  - Disabling the transmitter enables the remote end of the redundant link to also be in the operational Down state, which might be a requirement for third-party equipment when supporting redundancy over LAG.
  - Enabling the transmitter provides for a quick LAG failover in the event one of the non-redundant links in the LAG fail. This is particularly true when LACP has been enabled on the LAG, because it can take several seconds for LACP to converge on a link.
- Use the **auto-revert** keyword to instruct the redundant link to revert back to redundant mode when the failed link becomes active again.
- Example 1—Specifies that the Gigabit Ethernet interface in slot 4, port 0 is a redundant member interface

  host1(config-if)#**redundant-port gigabitEthernet 4/0**

- Example 2—Specifies that the Gigabit Ethernet interface in slot 1, port 1 is a redundant member interface with a packet sampling delay of 500 ms

  host1(config-if)#**redundant-port gigabitEthernet 1/1 packet-sampling delay 500**

- Use the **no** version to disable the redundant status of the member interface or disable the specified redundancy setting for the member.

redundant-port force-failover

- Use to force the specified member interface to fail over when more than one active member exists.
- Example

  host1(config)#**redundant-port gigabitEthernet 4/0 force-failover**

- There is no **no** version.

Monitoring 802.3ad Link Aggregation

This section explains how to use the **show** commands to display the characteristics and the configured settings for 802.3ad link aggregation.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

show interfaces lag

- Use to display information about a specified Ethernet member link in an IEEE 802.3ad link aggregation group (LAG) bundle.
- Specify either the Fast Ethernet or Gigabit Ethernet interface type when issuing this command:

  host1(config):**show interfaces *interfaceType interfaceSpecifier* lag**

- Field descriptions
  - *interfaceSpecifier*—Status of the hardware on this interface
    - Up—Hardware is operational
    - Down—Hardware is not operational
  - Administrative status—Operational state that you configured for this interface
  - Member—Membership status of the Ethernet link
  - LACP—Status of LACP configuration for the Ethernet link
    - active—Ethernet link participates in the protocol regardless of whether its Partner member link is set to active or passive LACP PDU participation
    - passive—Ethernet link transmits LACP PDUs only when it receives LACP PDUs from its Partner member link
  - mux state—Status of collecting and distributing at the Mux state machine
    - collecting/distributing—Ethernet link is actively collecting incoming frames and distributing outgoing frames
    - detached—Ethernet link is detached from the LAG bundle due to protocol changes or system constraints
    - waiting—Ethernet link is waiting to attach to a LAG bundle

- LACP state
  - active—Actor link actively particulates in LACP
  - passive—Actor link transmits LACP PDUs
  - timeout—Timeout control value; this value is not configurable and is set to long timeout (30 seconds)
  - aggregatable—Actor link can be aggregated
  - individual—Actor link cannot be aggregated; must operate as an individual link
  - in-sync—Actor link has joined the correct LAG bundle
  - out-of-sync—Actor link is unable to join the correct LAG bundle
  - collecting—Actor link is actively collecting incoming frames; if this field does not appear, the Actor link is not actively collecting incoming frames
  - distributing—Actor link is actively distributing outgoing frames; if this field does not appear, the Actor link is not actively distributing outgoing frames
  - defaulted—Actor link is using defaulted operational information about the Partner link that was administratively configured for Partner; if this field does not appear, the operational information about the Partner link has been received by the Actor link in an LACP PDU
  - expired—Actor link's receive machine is expired; if this field does not appear, the Actor link's receive machine is active
- port—Port number assigned to the Ethernet link by the Actor link
- priority—Priority assigned to this Ethernet link by the Actor link
- Key—Operational key value assigned to the Ethernet link by the Actor link
- System Priority—Priority assigned to the Ethernet link by the system
- System MAC Address—MAC address assigned to the Actor link
- Partner—Status of the Partner link
  - active— Partner link participates in the LACP
  - passive—Partner link transmits LACP PDUs
  - timeout—Timeout control value; short timeout or long timeout
  - aggregatable—Partner link can be aggregated
  - individual—Partner link cannot be aggregated
  - in-sync—Partner link has joined the correct LAG bundle
  - out-of-sync—Partner link has joined the incorrect LAG bundle
  - collecting—Partner link is actively collecting incoming frames; if this field does not appear, the Partner link is not actively collecting incoming frames
  - distributing—Partner link is actively distributing outgoing frames; if this field does not appear, the Partner link is not actively distributing outgoing frames

        - defaulted—Partner link is using defaulted operational information about the Partner link that was administratively configured for Partner; if this field does not appear, the operational information about the Partner link has been received by the Actor link in an LACP PDU
        - expired—Partner link's receive machine is expired; if this field does not appear, the Partner link's receive machine is active
        - port—Port number assigned to the Ethernet link by the Partner link
        - priority—Priority assigned to the Ethernet link by the Partner link
        - key—Operational key value assigned to the Ethernet link by the Partner link
        - age—Number of seconds since last LACP was received
        - System Priority—Priority assigned to the Ethernet link by the Partner link's system
        - System MAC Address—MAC address assigned to the Partner link by the system
    - LACP packets—Number of transmitted and received LACP packets
    - Marker Protocol request packets—Number of Marker Protocol packets requested to verify transmissions
    - Marker Protocol response packets—Number of Marker Protocol response packets that verified transmissions
    - Discarded—Number of invalid LACP packets
- Example

```
host1#show interfaces fastEthernet 4/0 lag
FastEthernet4/0 is Up, Administrative status is Up
 Member of Lag boston
 LACP passive, mux state collecting/distributing
 LACP state (0x3c) passive, long timeout, aggregatable, in-sync, collecting,
distributing
 port 0 priority 32768 key 8
 System Priority 32768 System MAC Address is 0090.1a40.2043
 Partner: state (0x3d) active, short timeout, aggregatable, in-sync,
collecting, distributing
  port 0 priority 32768 key 8 age 25
  System Priority 32768 System MAC Address is 0090.1a40.2043

 LACP packets: received 8, transmitted 7
 Marker Protocol request packets: received 0, transmitted 0
 Marker Protocol response packets: received 0, transmitted 0
 Discarded 0, unknown protocol received 0
```

show interfaces lag members

- Use to display information about the Ethernet member links in all IEEE 802.3ad link aggregation group (LAG) bundles configured on the router, or about the member links in a specified IEEE 802.3ad LAG bundle.
- Field descriptions
  - Lag—Name of the LAG bundle
  - Administrative status—Operational state that you configured for the LAG
  - Member-interface—Status of the member interface in the bundle
    - *Interface Specifier*—Status of the hardware on this interface (up or down)
    - LACP active—Ethernet link participates in the protocol regardless of whether its Partner member link is set to active or passive LACP PDU participation
    - LACP passive—Ethernet link transmits LACP PDUs only when it receives LACP PDUs from its Partner link
    - collecting/distributing—Ethernet link is actively collecting incoming frames and distributing outgoing frames
    - detached—Ethernet link is detached from the LAG bundle due to protocol changes or system constraints
    - waiting—Ethernet link is waiting to attach to a LAG bundle
- Example

```
host1#show interfaces boston lag members

Lag bostonBundle is Up, Administrative status is Up
   Member-interface FastEthernet0/0 is Up
      (LACP active, state collecting/distributing)
   Member-interface FastEthernet0/5 is Up
      (LACP active, state collecting/distributing)

Lag actonBundle is Up, Administrative status is Up
   Member-interface FastEthernet4/0 is Up
      (LACP passive, state collecting/distributing)
   Member-interface FastEthernet4/6 is Up
      (LACP passive, state collecting/distributing)
2 lag interfaces found
```

Chapter 7

Configuring Point-to-Point Protocol

This chapter describes how to configure a Point-to-Point Protocol (PPP) interface on E-series routers.

This chapter contains the following sections:

Overview

PPP provides a standard method for transporting multiprotocol datagrams over a point-to-point link. PPP uses the High-Speed Data Link Control (HDLC) protocol for its physical interface and provides a packet-oriented interface for the network-layer protocols.

Internet Protocol Control Protocol (IPCP) (which negotiates for transport of IP version 4 datagrams), IPv6CP (which negotiates for transport of IP version 6 datagrams), the OSI Network Layer Control Protocols (OSINLCPs), and Multiprotocol Label Switching (MPLS) run within PPP.

The router supports dynamic PPP interfaces. For details, see *Chapter 15, Configuring Dynamic Interfaces*.

Framing

The software restricts the use of the general HDLC protocol (RFC 1662) to unnumbered mode:

- HDLC address field is 0xFF (all stations)
- HDLC control field is 0x03 (to indicate unnumbered mode)

The router does not support the following framing features:

- Numbered mode (RFC 1663)
- Autodetection of encapsulation

Error Frames

The router relies on higher-layer protocols to recover from PPP data loss. All unrecognized protocol data units (PDUs) are discarded; however, statistics are maintained for packets dropped.

Link Control Protocol

PPP's Link Control Protocol (LCP) establishes a PPP link by negotiating with the PPP peer at the other end of a proposed connection. When two routers initialize a PPP dialogue, each router sends control packets to the peer. The control packets contain a list of LCP options and corresponding values that the sending peer uses to define its end of the link, such as the maximum receive unit (MRU).

LCP negotiations continue until the peers either converge (that is, reach an agreement about values for connection parameters) or abandon attempts to establish a connection.

If you configure a PPP interface without an IP interface or profile, the router negotiates LCP, but then terminates LCP after 2 to 3 minutes. Previously, the behavior in such a circumstance was to negotiate LCP and then leave LCP open.

For static PPP interfaces, whenever LCP achieves a stopped state because of termination, negotiation failure, or some other cause, it goes into passive mode and waits for the other side of the connection to restart the negotiation process. Once in passive mode, the router periodically attempts to negotiate with the other side according to an exponential timeout algorithm.

For static PPP interfaces, the router waits 15 seconds, attempts negotiation, waits 30 seconds if it fails, attempts negotiation, waits 60 seconds if it fails, and so on. The timeout periods are 15 seconds, 30 seconds, 60 seconds, 2 minutes, 4 minutes, 8 minutes, and 15 minutes. Once it reaches the 15-minute timeout, the router attempts negotiation every 15 minutes until successful. When LCP reaches the open state, the timer resets to 15 seconds.

Dynamic PPP interfaces are always torn down when LCP achieves a stopped state. For more information, see *Chapter 15, Configuring Dynamic Interfaces*.

LCP Negotiation Parameters

LCP can negotiate many PPP options, as follows:

- MRU size—Maximum receive unit size (always accepted).
- Magic number—Randomly generated number used to identify one end of a point-to-point connection. Each side negotiates its magic number, taking note of each other's magic number. If both sides discover that the magic numbers they are negotiating are the same, each side attempts to change its magic number. If they are not successful, and the magic numbers remain the same, the session terminates because of the loopback that is detected. Magic numbers are always accepted.

  By default, the router always attempts to negotiate a local magic number. The peer can also determine whether to negotiate its magic number—the peer magic number. The router always accepts a peer's attempt to negotiate its magic number.

  If the peer does not attempt to negotiate its magic number, you can configure the router to ignore a mismatch of the peer magic number and retain the PPP connection. For details, see *Validation of LCP Peer Magic Number* on page 224.
- Authentication—Requested if configured.
- Protocol-Field-Compression (PFC) and Address-and-Control-Field-Compression (ACFC)—Accepted, but never requested.
- Multilink PPP—Additional options can be negotiated when Multilink PPP is configured. See *Chapter 8, Configuring Multilink PPP*.
- Async-Control-Character-Map (ACCM—Supported by PPP when used with an L2TP Network Server (LNS). ACCM allows PPP to indirectly support asynchronous PPP connections tunneled via a third-party L2TP Access Concentrator (LAC). PPP on the router uses the ACCM configuration data as supplied by the LAC via proxy LCP. The router does not directly support asynchronous PPP connections and will not negotiate an ACCM option unless directed to do so by a third-party LAC.

PPP can also detect a loopback that occurs after LCP is negotiated, provided that:

- No loopback occurs during LCP negotiations.
- A loopback is introduced after LCP negotiation without forcing LCP renegotiation. (LCP is renegotiated if the lower layer goes down or if an LCP confReq is received from the other end.)

Validation of LCP Peer Magic Number

If the peer has not negotiated an LCP magic number, you can configure the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection.

Previously, the router terminated a PPP connection with a non-conforming peer when it received LCP echo request packets or LCP echo reply packets from the peer with a magic number that did not match the LCP peer magic number on the router. This is still the current default behavior if you do not explicitly configure the router to ignore the LCP peer magic number mismatch if the peer has not negotiated the magic number and retain the PPP connection.

Configuring the router to ignore the peer magic number mismatch and retain the PPP connection is useful if your network includes peers that send a non-null or invalid magic number in the LCP echo request and reply packets despite having not negotiated the magic number. In this situation, the router expects to receive a null magic number from the peer, and terminates the PPP connection unless you configure it to ignore the peer magic number mismatch and retain the connection.

To configure the router to ignore the LCP peer magic number mismatch and retain the PPP connection, use the **ppp magic-number ignore-mismatch** command from Interface Configuration mode or Subinterface Configuration mode. For more information, see **ppp magic-number ignore-mismatch** on page 237.

To verify configuration of LCP peer magic number validation on the router, you can use the **show ppp interface** command. For more information, see **show ppp interface** on page 243.

Keep the following points in mind when configuring the router to ignore the peer magic number mismatch and retain the PPP connection:

- If the peer negotiates the magic number but sends the router an LCP echo request or reply packet that contains a null or invalid magic number, the router strictly terminates the PPP connection. The router can ignore a mismatch of the LCP peer magic number only when the peer has not negotiated the magic number.

- Using the **ppp magic-number disable** command to disable negotiation of the magic number on the router does not affect validation of the peer magic number. When you issue the **ppp magic-number disable** command, the router sets only the local magic number to null, but does not change or validate the peer magic number. (For more information, see **ppp magic-number disable** on page 236.)

You can also configure validation of the LCP peer magic number for static MLPPP interfaces, dynamic PPP interfaces, and dynamic MLPPP interfaces. For more information about configuring static MLPPP interfaces, see *Chapter 8, Configuring Multilink PPP*. For more information about using profiles to configure dynamic PPP and dynamic MLPPP interfaces, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

B-RAS Support

Broadband Remote Access Server (B-RAS) is an application that aggregates the output from digital subscriber line access multiplexers (DSLAMs). B-RAS provides user PPP sessions and PPP session termination and routes traffic onto the backbone. See *JUNOSe Broadband Access Configuration Guide, Chapter 1, Configuring Remote Access* for details on B-RAS.

The router provides an enhanced version of PPP to accommodate B-RAS with the following features:

- Internet Protocol Control Protocol (IPCP) extensions for Windows Internet Name Service (WINS) and Domain Name System (DNS) name server addresses
- Password Authentication Protocol (PAP)
- Challenge Handshake Authentication Protocol (CHAP)
- Keepalive timeout
- Session timeout
- Inactivity timeout
- Accounting

Authentication

The router acts as an authenticator. It demands authentication from a remote PPP peer but refuses to authenticate itself.

Rate Limiting for PPP Control Packets

The router implements rate limiting for PPP control packets to protect the corresponding PPP interface from denial-of-service (DoS) attacks. The interface discards control packets when the rate of control packets received exceeds the rate limit for PPP interfaces.

A PPP interface has a rate limit control that is non-configurable and always in effect; the rate limit is the same for all PPP interfaces. In addition, each interface instance maintains its own state and statistics counters for tracking the rate. The rate limit for PPP control packets is approximately 10 packets per second.

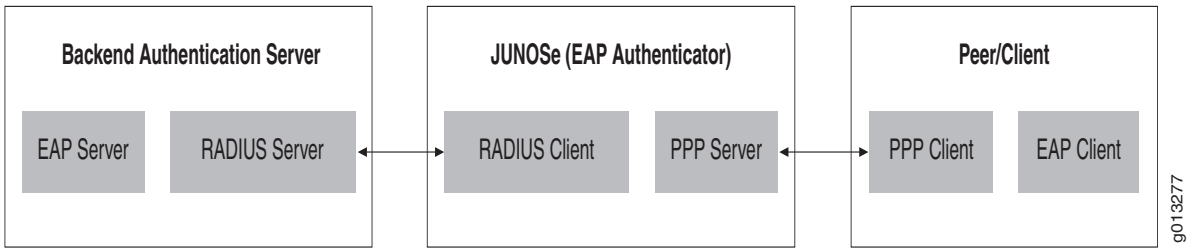
For a PPP interface, the router increments the discards counter in the **show ppp interface** command display to track the number of PPP control packets discarded on receipt (in) or discarded before they were transmitted (out) on this interface.

For examples of the **show ppp interface** command display, see **show ppp interface** on page 243.

Extensible Authentication Protocol

The JUNOSe software supports Extensible Authentication Protocol (EAP) for authenticating a peer before allowing network layer protocols to transmit over the link. EAP supports multiple authentication methods, including EAP-TLS and EAP-MD5-Challenge. The EAP server and the peer negotiate the specific authentication method to be used. Figure 30 illustrates the three components required for EAP: an EAP authenticator, an EAP server, and an EAP client.

Figure 30: Authentication with EAP



After LCP negotiation, JUNOSe starts the EAP negotiation process by initiating an identity exchange with the EAP client on the peer. The router sends an EAP identity request packet to the peer, which replies with an EAP identity response packet. After this exchange, the E-series router acts only as a pass-through device, enabling the EAP server residing on the backend authentication server to select and negotiate the particular EAP authentication method directly with the EAP client on the peer.

The JUNOSe software forwards or discards packets received from the backend authentication router and the peer depending on the identifying code contained in the packet.

The E-series router forwards:

- Packets received from the peer with a Response code
- Packets received from the backend authentication server with a Request, Success, or Failure code

The E-series router discards:

- Packets received from the peer with a Request, Success, or Failure code
- Packets received from the backend authentication server with a Response code

The JUNOSe software determines the outcome of the authentication based only on the Accept or Reject indication sent by the RADIUS server

EAP Types

The JUNOSe software has been qualified to work with the EAP authentication methods—known as EAP types—described in Table 12. Other EAP authentication methods have not been qualified with the JUNOSe software.

Table 12: Supported EAP Types

EAP Type	Behavior
1—Identity	When LCP negotiation completes, PPP sends an initial EAP identity request packet to the peer. The EAP identity response packet received from the peer is forwarded to AAA. AAA forwards the response as an Access-Request to the RADIUS server hosted on the backend authentication server.
2—Notification	The JUNOSe software forwards Notification requests from the backend authentication server to the peer and Notification responses from the peer to the server. The JUNOSe software does not initiate any Notification requests or responses.
3—NAK	The JUNOSe software forwards the NAKs received from the peer to the backend authentication server.
4—MD5-Challenge	The JUNOSe software acts as a pass-through for the EAP-MD5-Challenge negotiated between the peer and backend authentication server.
13—TLS	The JUNOSe software acts as a pass-through for the EAP-TLS negotiated between the peer and backend authentication server.

EAP Packet Retransmission

PPP retransmits the EAP request packets to the peer. The RADIUS client retransmits the EAP response packets to the RADIUS Server. The request packets to the peer are governed by nonconfigurable values for retransmission attempts and interval. The configuration of the RADIUS client determines retransmission values for response packets to the RADIUS server. The retransmission values are as follows:

- PPP makes five attempts to retransmit an EAP request before the authentication attempt is terminated. You cannot configure the number of retransmission attempts.

- When an EAP request is transmitted, a timer is started with a nonconfigurable retransmission interval value of 3 seconds. When the timer expires, the EAP request is retransmitted.

In some cases, you might want a longer retransmission interval. For example, you might need to accommodate the additional time required by a user to enter information or scan a fingerprint or retina. RADIUS can instruct the JUNOSe software to wait longer by passing an appropriate Session-Timeout attribute in the RADIUS Access-Challenge packet. This retransmission interval value applies only to the EAP request packet present in the RADIUS Access-Challenge packet.

The Session-Timeout attribute value overrides the default retransmission interval value, up to a maximum of 30 seconds. If RADIUS recommends a greater value, then PPP resets it back to 30 seconds in order to avoid longer or infinite delays.

EAP Behavior in an L2TP Environment

EAP behavior in an L2TP environment varies depending on whether the router acts as a LAC or an LNS,

When the E-series Router Acts as a LAC

When PPP forwards an EAP identity response packet to AAA, AAA might be configured to return a tunnel response upon successful validation of the packet. You can use AAA domain maps, a AAA profile, or both to force such tunneling.

On an LAC, PPP forwards the PPP EAP authentication information to the LNS during the establishment of the L2TP session. This authentication information consists of the EAP type, the data appropriate to the type (such as a username) contained in the EAP identity response packet, and the identifier of the EAP identity response packet. If the LNS trusts the LAC, then the LNS uses this authentication information to resume the EAP negotiation where the LAC left off.

L2TP on an LAC forwards the PPP EAP authentication information in the Proxy Authen AVPs as described in L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration).

When the E-series Router Acts as an LNS

PPP on an LNS resumes the EAP negotiation operation by detecting the presence of EAP information in the proxy authentication data supplied by L2TP. PPP reconstructs the EAP identity response packet from the proxy authentication data and forwards it to AAA.

L2TP on an LNS processes the received Proxy Authen AVPs as described in L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration).

Limitations

EAP is subject to internal limits. When the E-series router acts as a pass-through between the backend authentication server and the peer, EAP packets traverse the controllers within the router. The size of EAP packets and fragments tends to be larger than the buffer exchange limit—1450 bytes—between the controllers. This intercontroller buffer exchange limit is tuned for the optimal system performance and scalability; also, when stacked over L2TP on LNS, it prevents PPP control packets from causing IP fragmentation and reassembly on the Ethernet downlink. Hence, if EAP is configured as a PPP authentication protocol, then EAP packet or fragment size is affected by the intercontroller buffer exchange limit as follows:

- The MRU value advertised by JUNOSe in the LCP request packet takes the lowest of the following values:
  - the lower layer MRU minus the PPP overhead
  - the configured MRU
  - 1450 bytes
- The MTU value is initialized by JUNOSe to the lowest of the following values:
  - the lower layer MTU minus the PPP overhead
  - the peer MRU
  - 1450 bytes

The MTU value is passed to RADIUS in an Access-Request packet by means of the Framed-Mtu attribute.

Performance

When EAP is configured on the router, it affects the performance and scalability of PPP in terms of round-trip packet exchanges, negotiations, EAP server requirements, and EAP client requirements. For information on the number of PPP interfaces supported with EAP, see the *Link Layer Maximums* tables in *Appendix A, System Maximums*, of the current *JUNOSe Release Notes*.

- Performance depends on the number of packets exchanged during the negotiation. When the number of packets exchanged increases—that is, when the number of round-trips increases—it takes longer to finish the interface negotiation. System resources are locked for a longer duration. As a result it takes longer to bring up all the interfaces.

  The number of round-trip message exchanges varies with the EAP authentication method. When no retransmission of packets takes place and there is no fragmentation, PAP and CHAP require one round-trip, EAP-MD5-Challenge requires two round-trips, and EAP-TLS requires four round-trips.

  Retransmission increases the number of round-trips. When the negotiated EAP authentication method requires fragmentation, such as for the exchange of large certificate chains, then the number of round-trips increases.

- The number of simultaneous EAP negotiations is limited to 50 because of resource limitations. Consequently, the time required to bring up interfaces when you configure EAP authentication is longer than when you specify PAP or CHAP authentication.

- EAP authentication methods fragment packets when the EAP packet size is greater than the link MTU. The EAP server must fragment the EAP packet to the size of the Framed-Mtu attribute contained in the RADIUS Access-Request packet.

  If the server fragments the packet to a larger size than specified by the attribute, then JUNOSe drops the packet, because the E-series router acts as a pass-through device and is not involved in the authentication method's fragmentation and reassembly mechanisms.

  On the other hand, if the EAP server fragments the EAP packet to a smaller size than specified by the attribute, then performance decreases because of the increased number of smaller packets that must be exchanged.

- The EAP client on the peer must fragment the EAP packets to the size of the link MTU on the E-series router. When it does not do so, performance can be affected.

Platform Considerations

You can configure PPP interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support PPP interfaces on ERX-14xx models, ERX-7xx 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 PPP.

For information about the modules that support PPP interfaces 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 PPP.

Interface Specifiers

Some of the configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the physical interface on which you want to configure PPP. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router. n

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about the PPP protocol, consult the following resources:

- L2TP Proxy Authenticate Extensions for EAP—draft-ietf-l2tpext-proxy-authen-ext-eap-01.txt (December 2006 expiration)
- RFC 1332—The PPP Internet Protocol Control Protocol (IPCP) (May 1992)
- RFC 1661—The Point-to-Point Protocol (PPP) (July 1994)
- RFC 1662—PPP in HDLC-like Framing (July 1994)
- RFC 1877—PPP Internet Protocol Control Protocol Extensions for Name Server Addresses (December 1995)
- RFC 1994—PPP Challenge Handshake Authentication Protocol (CHAP) (August 1996)
- RFC 2153—PPP Vendor Extensions (May 1997)
- RFC 2246—The TLS Protocol Version 1.0 (January 1999)
- RFC 2615—PPP over SONET/SDH (June 1999)
- RFC 2716—PPP EAP TLS Authentication Protocol (October 1999)
- RFC 3032—MPLS Label Stack Encoding (January 2001)
- RFC 3579—RADIUS EAP (September 2003)
- RFC 3748—Extensible Authentication Protocol (EAP) (June 2004)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Before You Configure PPP

Before you configure a PPP interface, configure the interface or tunnel over which PPP traffic will flow. See the following chapters:

- *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 6, Managing Tunnel-Service and IPSec-Service Interfaces*
- *Chapter 1, Configuring ATM*
- *Chapter 9, Configuring Packet over SONET*
- *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*

The procedures described in this chapter assume that a physical interface has been configured.

Configuration Tasks

The following procedure is an example of a PPP configuration on a serial interface. These steps are mandatory unless otherwise noted.

1. From Global Configuration mode, specify the physical interface on which you want to configure PPP.

   host1(config)#**interface serial 3/0:2/5**

2. Specify PPP as the encapsulation method (data-link protocol) on the interface.

   host1(config-if)#**encapsulation ppp**

3. Assign an IP address and subnet mask for the interface.

   host1(config-if)#**ip address 192.168.22.10 255.255.255.0**

4. Verify that your configuration changes are correct.

   host1#**show ppp interface serial 3/0:2/5 config**

encapsulation ppp

- Use to configure PPP as the encapsulation method.
- Example

  host1(config-if)#**encapsulation ppp**

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

interface atm

- Use to specify a previously configured ATM interface on which you want to configure PPP.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config-if)#**interface atm 9/1.1**
  host1(config-if)#**interface atm 5/0/1.1**

- Use the **no** version to disable or remove the subinterface or the logical interface.

interface pos

- Use to specify a previously configured packet over SONET (POS) interface on which you want to configure PPP.
- To specify a POS interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface

- To specify a POS interface for E120 and E320 routers, use the *slot/adapter/port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- For more information about modules that support POS interfaces, see *JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*.
- Examples

  host1(config-if)#**interface pos 0/1**
  host1(config-if)#**interface pos 5/0/0**

- Use the **no** version to remove the POS interface.

interface serial

- Use to specify a serial interface in the *slot/port:channel/subchannel* format by selecting a previously configured physical interface on which you want to configure PPP.
  - *slot*—Refers to a router chassis slot.
  - *port*—Refers to a CT3, T3, or E3 module I/O port.
  - *channel*—Refers to a T1 (DS1) channel.
  - *subchannel*—Represents a set of DS0 subchannels.
- Example

  host1(config)#**interface serial 3/0:2/5**

- Use the **no** version to disable or remove the subinterface or the logical interface.

ip address

- Use to assign an IP address and subnet mask for a PPP interface.
- Example

  host1(config-if)#**ip address 192.168.22.10 255.255.255.0**

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

Optional Configuration Tasks

You can perform the following optional PPP configuration tasks:

- Add a text description or alias to a PPP interface.
- Configure the IPCP netmask option (option 0x90).
- Specify the keepalive timeout value.
- Disable magic numbers.
- Control validation of the LCP peer magic number when the peer has not negotiated an LCP magic number.
- Specify the maximum receive units.
- Configure passive mode.
- Configure name server addressing.
- Stop or restart a PPP session.
- Configure PPP authentication.

ppp description

- Use to assign a text description or alias to a static PPP interface.
- Example

  host1(config-if)#**ppp description pah8999**

- Use the **no** version to remove the description.

ppp ipcp netmask

- Use to specify the IPCP netmask option (option 0x90) for each PPP interface. By default, the IPCP netmask option is disabled on the interface.
- The IPCP netmask option is a nonstandard option that enables a peer to request the netmask associated with the assigned IP address.
- The netmask can be specified via RADIUS attribute 9, Framed-Ip-Netmask. If the netmask is 255.255.255.255, the option is not negotiated. See the **radius ignore framed-ip-netmask** command.
- You can enable the IPCP netmask option either in a profile or on a static interface.
- Example

  host1(config-subif)#**ppp ipcp netmask**

- Use the **no** version to disable the IPCP netmask option on the interface.

ppp keepalive

- Use to specify the keepalive timeout value.
- There are two keepalive modes of operation: high-density mode and low-density mode.
  - High-density keepalive mode is automatically selected if PPP is layered over ATM, L2TP, or PPPoE.
  - Low-density keepalive mode is selected if PPP is layered over HDLC. Keepalive mode selection is made per interface.
- High-density mode—This mode is also known as smart keepalive. When the keepalive timer expires, the interface first verifies whether any frames were received from the peer in the prior keepalive timeout interval. If so, the interface does not send an LCP echo request (keepalive). Keepalive packets are sent only if the peer is silent (that is, no traffic was received from the peer during the previous keepalive timeout interval). If both sides are configured with keepalive, receipt of an LCP echo request by one end suppresses the transmission of an LCP echo request by that end. Smart keepalive is disabled when the keepalive timeout value is at least 60 seconds, even when in high-density mode. Smart keepalive is always disabled when in low-density mode. This mode suppresses transmission of unnecessary LCP echo requests.
- For high-density keepalive mode, the range is 30–64800 seconds. The default value is 30 seconds.
- Low-density mode—When the keepalive timer expires, the interface *always* sends an LCP echo request, regardless of whether the peer is silent.
- For low-density keepalive mode, the range is 1–64800 seconds for POS uplink interfaces, and 10–64800 seconds for all other HDLC interfaces. The default value for all interfaces is 30 seconds.
- If the keepalive interval is 30 seconds, a failed link is detected between 90 and 120 seconds after failure.
- Use **ppp keepalive** without a value to restore the default, 30 seconds.
- Example

  host1(config-if)#**ppp keepalive 50**

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

ppp magic-number disable

- Use to disable negotiation of the local magic number.
- Issuing this command prevents the router from detecting loopback configurations.
- Example

  host1(config-if)#**ppp magic-number disable**

- Use the **no** version to restore negotiation of the local magic number.

ppp magic-number ignore-mismatch

- Use to cause the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.
- For more information about using this command, see *Validation of LCP Peer Magic Number* on page 224.
- Example

  host1(config-if)#**ppp magic-number ignore-mismatch**

- Use the **no** version to restore the default behavior, in which the router terminates the PPP connection if it detects an LCP peer magic number mismatch.

ppp mru

- Use to control the negotiation of the maximum receive unit (MRU).
- Specify the number of bytes, in the range 64–65535.
- We recommend you coordinate this value with the network administrator on the other end of the line.
- If the value configured for the PPP MRU is greater than the value of the lower-layer MRU minus the PPP header length, the router logs a warning message and uses the lesser of the configured MRU value or the lower-layer MRU value minus the PPP header length to negotiate the local MRU.
- If the value configured for the PPP MRU conflicts with a similar value configured for another protocol, such as the MTU value for PPPoE, the router uses the lesser of the two values.
- Example

  host1(config-if)#**ppp mru 576**

- Use the **no** version to restore the default value, which causes PPP to use the lower-layer MRU minus the PPP header length as the MRU value.

ppp passive-mode

- Use to force a static or dynamic PPP interface into passive mode before LCP negotiation begins, for a period of one second. This delay enables slow clients to start up and initiate the LCP negotiation.
- Example

  host1(config-if)#**ppp passive-mode**

- Use the **no** version to disable passive mode.

ppp peer

- Use to resolve conflicts when the router and the PPP peer have the primary and secondary DNS and WINS name server addresses configured with different values.
- By default, the DNS and WINS addresses configured on the router take precedence.

- Use the **dns** keyword or the **wins** keyword to configure which PPP peer address takes precedence. This command has no effect unless both routers have the address configured and the address is in conflict. If the PPP peer has the address and the router does not, the peer always supplies the address regardless of how you have configured the PPP peer.
- Example

  host1(config-if)#**ppp peer dns**

- Use the **no** version when you want the router to take precedence during setup negotiations between the router and the peer. If the IP addresses that the peer sends to the router differ from the ones configured on your router, the router returns the values that you configured as the correct values to the peer.

ppp shutdown
ppp shutdown ip
ppp shutdown ipv6
ppp shutdown mpls
ppp shutdown osi

- Use to terminate a PPP session.
- To administratively disable the interface, use the **ppp shutdown** command.
- To administratively disable IPCP, use the **ppp shutdown ip** command.
- To administratively disable IPv6CP, use the **ppp shutdown ipv6** command.
- To administratively disable MPLS, use the **ppp shutdown mpls** command.
- To administratively disable OSINLCP, use the **ppp shutdown osi** command.
- All PPP sessions are enabled by default.
- Example

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

- Use the **no** version to restart a disabled session.

Configuring PPP Authentication

Perform the following optional tasks to configure PPP authentication:

- Specify one or more PPP authentication types, and select an authentication virtual router context.
- Specify the CHAP challenge length.
- Specify the maximum number of retries.

NOTE: The JUNOSe software's PPP application accepts null usernames during PAP and CHAP authentication. When the PPP application receives an authentication request that includes a null username, PPP passes the request to AAA. To take advantage of this feature, configure your authentication server to support the use of null usernames.

ppp authentication

- Use to request authentication from a PPP peer and set the authentication method.
- To specify the name of a virtual router (VR) to be used as the authentication VR context, use the **virtual-router** keyword. Keep the following points in mind when you use the **ppp authentication virtual-router** command:
  - When you specify a VR in the **ppp authentication** command, AAA does not query the domain map for the assigned VR context. Instead, AAA uses the VR specified in the **ppp authentication** command as the authentication VR context and issues the authentication request to the authentication server in the assigned VR context.
  - If you specify the default VR as the authentication VR context, AAA loosely binds the user to the default VR. This means that RADIUS *can override* the default VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies the default VR, AAA returns either the default VR or the VR specified by RADIUS.
  - If you specify a VR other than the default VR as the authentication VR, AAA tightly binds the user to the specified VR. This means that RADIUS *cannot override* the specified VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies a nondefault VR, AAA returns the specified VR.
- The router supports the MD5 authentication algorithm for CHAP authentication.
- You can specify one or more authentication protocols in order of preference. If the peer router refuses the first choice, then the local router requests the next authentication protocol, if specified. If the peer refuses that protocol, then the local router requests the third protocol, if specified. If the peer refuses all specified authentication protocols, then the local router terminates the session.
- Example 1—Specifies the order of preference for the primary authentication protocol

  host1(config-if)#**ppp authentication pap chap eap**

  The router requests the use of PAP as the authentication protocol (because it appears first in the command line). If the peer refuses to use PAP, the router requests the CHAP protocol. If the peer refuses to use CHAP, the router requests the EAP protocol. If the peer refuses to negotiate authentication, the router terminates the PPP session.

- Example 2—Specifies a virtual router for the authentication virtual router context

  host1(config-if)#**ppp authentication virtual-router boston pap chap**

  This command is available in static configurations and in profiles.

- Example 3—Configures only EAP on a static PPP interface

  host1(config)#**interface atm 3/2.100**
  host1(config-subif)#**ppp authentication eap**

- Example 4—Configures EAP or PAP on a static PPP interface

  host1(config)#**interface atm 3/2.100**
  host1(config-subif)#**ppp authentication eap pap**

  EAP negotiation is attempted first. If PPP receives a NAK from the peer in response to the EAP request, then PAP is attempted. If PAP is also rejected, then PPP terminates the session.

- Example 5—Configures only EAP on a dynamic PPP interface

  host1(config)#**profile ppptest**
  host1(config-profile)#**ppp authentication eap**

- Example 6—Configures EAP or CHAP or PAP on a dynamic PPP interface

  host1(config)#**profile ppptest**
  host1(config-profile)#**ppp authentication eap chap pap**

  In this example, the router first attempts EAP negotiation. If PPP receives a NAK from the peer in response to the EAP request, then the router attempts CHAP negotiation. If PPP receives a NAK from the peer in response to the CHAP request, then the router attempts PAP negotiation. If PAP is also rejected, then PPP terminates the session.

- Use the **no** version to specify that the router does not require authentication.

ppp chap-challenge-length

- Use to modify the length of the CHAP challenge by specifying the allowable minimum length and maximum length.
- Specify the minimum and maximum lengths in bytes in the range 8–63.

CAUTION: Do *not* decrease the range. Increasing the range is acceptable, provided that you do not lower the minimum to do so. The recommended minimum is 16. A longer challenge and a more unpredictable challenge length provide a higher level of security.

- The maximum length must be greater than or equal to the minimum length.
- Example

  host1(config-if)#**ppp chap-challenge-length 24 28**

- Use the **no** version to restore the default minimum (16 bytes) and default maximum (32 bytes).

ppp max-bad-auth

- Use to specify the maximum number of authentication retries the router allows before terminating a PPP session
- This value applies to PAP and CHAP authentication.
- The range is 0–7. The default is 0, which indicates that no retries are allowed.

- Example

  host1(config-if)#**ppp max-bad-auth 3**

- Use the **no** version to return the number of retries to the default, 0.

PPP Accounting Statistics

The JUNOSe software begins the collection of accounting statistics for terminated PPP sessions following, but not including, authentication acknowledgement from the E-series router. The acknowledgment is either a CHAP success or PAP acknowledgement packet. All subsequent traffic is counted up the point that PPP at the router terminates the subscriber's session. The statistics are reported in the following RADIUS attributes:

Attribute Number	Attribute Name
42	Acct-Input-Octets
43	Acct-Output-Octets
47	Acct-Input-Packets
48	Acct-Output-Packets

PPP session termination can be initiated through a number of mechanisms: PPP shutdown at the client or router interface, subscriber logout at the router (by means of the **logout subscriber** command), lower layer down events, and silent client termination.

The following rules apply to all termination scenarios:

- Accounting statistics reported in RADIUS octet counts (Acct-Input-Octets and Acct-Output-Octets) for terminated PPP customers include the following data:
  - All upper layer control traffic, including IPCP, IPCPv6, OSICP, and MPLSNCP
  - All data traffic, including IP, IPv6, MPLS, and OSI
  - All PPP LCP echo requests and responses following authentication
  - Other PPP LCP packets following the PAP or CHAP acknowledgment
  - Retransmits of the PAP or CHAP traffic

- PPP accounting statistics reported in RADIUS octet counts (Acct-Input-Octets and Acct-Output-Octets) exclude the following data:
  - PPP traffic prior to completion of authentication
  - PPP LCP terminate-request or terminate-acknowledgement packets
  - PPPoE padding for PPP control and data packets
- Accounting statistics reported in RADIUS packet counts (Acct-Input-Packets and Acct-Output-Packets) for terminated PPP customers are based on packets delivered to or received from the upper transport layer: IP, IPv6, MPLS, and OSI.

For information on accounting statistics for tunneled PPP sessions, see *PPP Accounting Statistics* in *JUNOSe Broadband Access Configuration Guide, Chapter 13, Configuring an L2TP LNS*.

Monitoring PPP Interfaces

Use the following versions of the **show ppp interface** command to monitor PPP interfaces:

- **show ppp interface**
- **show ppp interface summary**

You can set a statistics baseline for PPP interfaces using the **baseline ppp** commands. Use the optional **delta** keyword with PPP **show** commands to show baselined 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. Refer to *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

baseline ppp interface

- Use to establish a baseline for PPP statistics on an interface.
- The router implements the baseline by reading and storing the statistics at the time the baseline is set, then subtracting this baseline whenever baseline-related statistics are retrieved.
- Use the optional **delta** keyword with PPP **show** commands to show baselined statistics.

- Examples

  host1#**baseline ppp interface atm 3/3.20**
  host1#**baseline ppp interface atm 3/0/3.20**

- There is no **no** version.

show ppp interface

- Use to display selective PPP interface information.
- You can filter the command display for characteristics of particular interest, such as interface type, data type, configured protocol, or interface state.
- Field descriptions
  - PPP interface—Interface type, interface specifier, and status (up, down, lowerDown, not present, passive, or tunnel). For more information about specifying the physical interface on which you want to configure PPP, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - Interface alias—Alias or description of the PPP interface
  - Interface administrative status—Indicates whether the interface is administratively enabled (open), meaning that the **no ppp shutdown** command is operational; or administratively disabled (closed), which means that the **ppp shutdown** command is operational
  - Configured network protocol—Indicates the network protocol configured on the interface
  - Baseline status—Indicates whether a statistics baseline is set
  - Interface statistics
    - packets—Number of packets received (in) or transmitted (out) on the interface
    - octets—Number of octets received (in) or transmitted (out) on the interface
    - errors—Number of errors received (in) or transmitted (out) on the interface
    - discards—Number of packets discarded on receipt (in) or discarded before they were transmitted (out); for more information about the discards counter, see *Rate Limiting for PPP Control Packets* on page 225
  - IPCP protocol configuration
    - configured—IPCP is configured on this interface (true or false)
    - administrative-status—IPCP administrative status (open or closed)
    - ip-address—Address to be used for negotiation of the local IP address option
    - dns-precedence—Used to resolve conflicts during negotiation of DNS addresses; “local” indicates that the local side takes precedence and the **no ppp peer dns** command is operative; “peer” indicates that the remote side takes precedence and the **ppp peer dns** command is operative

- wins-precedence—Used to resolve conflicts during negotiation of WINS addresses; “local” indicates that the local side takes precedence and the **no ppp peer wins** command is operative; “peer” indicates that the remote side takes precedence and the **ppp peer wins** command is operative
- ipcp-netmask-option—Controls negotiation of the IPCP netmask option; disabled = do not negotiate, enabled = negotiate

- IPV6CP protocol configuration
  - configured—IPv6CP is configured on this interface (true or false)
  - administrative-status—IPv6CP administrative status (open or closed)
  - ipv6-interfaceId—Address to be used for negotiation of local IPv6 address option
- IPCP protocol status
  - operational-status—IPCP operational status (up, down, not present, or not present no resources)
  - terminate-reason—Reason for termination of IPCP service
- IPV6CP protocol status
  - operational-status—IPv6CP operational status (up, down, not present, or not present no resources)
  - terminate-reason—Reason for termination of IPv6CP service
- IPCP negotiated options—Shows the following negotiated addresses for the local and remote (peer) side of the link
  - ip-address—IP address
  - ip-address-mask—IP address mask
  - primary-dns-address—Primary DNS address
  - secondary-dns-address—Secondary DNS address
  - primary-wins-address—Primary WINS address
  - secondary-wins-address—Secondary WINS address

NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.

- OSINLCP protocol configuration
  - configured—OSINLCP is configured on this interface (true or false)
  - administrative-status—OSINLCP administrative status (open or closed)
- OSINLCP protocol status
  - operational-status—OSINLCP operational status (up, down, not present, or not present no resources)
  - terminate-reason—Reason for termination of OSINLCP service

- OSINLCP negotiated options
  - npdu-alignment—Negotiated NPDU alignment for the local and remote (peer) side of the link

NOTE: The command displays a value of "none" for any negotiated option parameters if the option was not negotiated.

- MPLSNLCP protocol configuration
  - configured—MPLSNLCP is configured on this interface (true or false)
  - administrative-status—MPLSNLCP administrative status (open or closed)
- MPLSNLCP protocol status
  - operational-status—MPLSNLCP operational status (up, down, not present, or not present no resources)
  - terminate-reason—Reason for termination of MPLSNLCP service
- MPLSNLCP negotiated options
  - npdu-alignment—Negotiated NPDU alignment for the local and remote (peer) side of the link

NOTE: The command displays a value of "none" for any negotiated option parameters if the option was not negotiated.

- LCP protocol configuration
  - max-receive-unit—Controls negotiation of the local MRU option; "use lower layer" indicates that the MRU of the layer below PPP defines the MRU to be negotiated; "disabled" indicates that the MRU option is not to be negotiated. A numeric value indicates the MRU value to be negotiated
  - authentication—Controls the negotiation of the local authentication option; "none" indicates do not negotiate; "chap" indicates negotiate chap; "pap" indicates negotiate pap; "chap/pap" indicates negotiate chap and, if it is rejected, negotiate pap; "pap/chap" indicates negotiate pap and, if it is rejected, negotiate chap.
  - magic-number—Controls whether the local magic number is negotiated: enabled (negotiate), or disabled (do not negotiate)
  - magic-number-mismatch—Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
  - keepalive-timer—Rate of LCP echo requests
  - restart-timer—Retry frequency during LCP, IPCP, OSINLCP, and MPLS negotiations
  - max-terminate—Maximum number of terminate requests
  - max-configure—Maximum number of configure requests

- max-failure—Maximum number of configure NAKs
- passive-mode—Forces a PPP interface into a passive mode before LCP negotiation begins; “disabled” means do not wait for peer; “enabled” means wait for peer to initiate negotiation

- LCP protocol status
  - link-status—Overall status of LCP negotiations, including the following states: Initial (idle), Starting (ready to negotiate), Authenticate (authenticating), and Network (LCP is up)
- LCP negotiated options—Shows the following negotiated values for the local and remote (peer) side of the link:
  - max-receive unit—Maximum receive unit, in octets
  - authentication—Authentication method (none, pap, or chap)
  - magic-number—Magic number
  - pfc—PFC (none or enabled)
  - acfc—ACFC (none or enabled)

NOTE: The command displays a value of “none” for any negotiated option parameters if the option was not negotiated.

- LCP Endpoint Discriminator options
  - local discriminator class—Endpoint discriminator type, format, and address space for the local and remote (peer) router
  - local endpoint discriminator—Endpoint discriminator value for the local router within the specified class
  - peer discriminator class—Endpoint discriminator type, format, and address space for the remote router
  - peer endpoint discriminator—Endpoint discriminator value for the remote router within the specified class
- LCP protocol statistics—Shows the following statistics for the life of the interface (since system boot or interface creation, whichever is later)
  - in-keepalive-requests—Number of received keepalive requests (LCP Echo Requests)
  - out-keepalive-requests—Number of transmitted keepalive requests
  - in-keepalive-replies—Number of received keepalive replies
  - out-keepalive-replies—Number of transmitted keepalive replies
  - keepalive-failures—Number of keepalive failures reported on the interface
- Authentication configuration
  - authenticate-retry—Maximum number of authentication retries configured using the **ppp max-bad-auth** command
  - authentication-router—Virtual router for the authentication virtual router context

- Authentication status
  - grant—Authentication status (true = access granted, false = access not granted)
  - session-timeout—Session timeout, in seconds; session is terminated at expiration
  - inactivity-timeout—Inactivity timeout, in seconds; session is terminated if it is not active for specified timeout
  - accounting-timeout—Accounting timeout in seconds; frequency of accounting updates to the authentication server
  - peer-ip-address—IP address to be used in negotiation of peer IP address
  - peer-ip-address-mask—IP address mask to be used in negotiation of the peer IP address mask
  - peer-primary-dns-address—IP address to be used in negotiation of the peer primary DNS address
  - peer-secondary-dns-address—IP address to be used in negotiation of the peer secondary DNS address
  - peer-primary-wins-address—IP address to be used in negotiation of the peer primary WINS address
  - peer-secondary-wins-address—IP address to be used in negotiation of the peer secondary WINS address

NOTE: The command displays the authentication status as "none" for any parameters not provided by the authentication server.

- Authentication statistics—Shows statistics accumulated since the session was established
  - up-time—Time the session has been up, in seconds
  - in-octets—Number of octets received on the interface
  - out-octets—Number of octets transmitted out the interface
  - in-packets—Number of packets received on the interface
  - out-packets—Number of packets transmitted out the interface
- PAP protocol configuration
  - request-timeout—Maximum time, in seconds, to wait for an authentication request packet
- CHAP protocol configuration
  - name—Name toM be used in challenge packets
  - challenge-retry—Maximum number of challenge packets to be transmitted
  - challenge-timeout—Frequency, in seconds, of challenge packet retransmission
  - minimum-challenge-length—Minimum length of challenge packet

- maximum-challenge-length—Maximum length of challenge packet; the size of the challenge used for each challenge packet is a random number between minimum-challenge-length and maximum-challenge-length
- minimum-rechallenge-timeout—Minimum time, in seconds, before initiating a rechallenge to peer
- maximum-rechallenge-timeout—Maximum time, in seconds, before initiating a rechallenge to peer; the actual time before a rechallenge is a random number between minimum-rechallenge-timeout and maximum-rechallenge-timeout

- If the operational status is down for a specific interface, one of the following termination reasons might appear in parentheses:
  - administrative disable—Interface has been administratively disabled, which means that the **ppp shutdown** command is in effect. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.
  - administrative logout—Interface has been administratively logged out, which means that the **logout subscriber** command has been issued. This applies to an interface only.
  - no upper interface—No upper layer is configured. This applies to an interface only.
  - authentication failure—Authentication is required and has failed. This applies to an interface only.
  - no local *xxx*—local option, *xxx,* is required and could not be negotiated (for example, IP address). This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.
  - no peer *xxx*—Remote peer option, *xxx*, is required and could not be negotiated (for example, authentication). This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.
  - keepalive drop count exceeded—Keepalive drop count has been exceeded. This applies to an interface only.
  - session timeout—Session timeout period has expired. This applies to an interface only.
  - inactivity timeout—Inactivity timeout period has expired. This applies to an interface only.
  - address lease expired—Address lease period has expired. This applies to an interface only.
  - not configured—Protocol is not configured on the interface. This applies to IPCP, IPv6CP, OSINLCP, and MPLS.
  - link down—Link is down and the protocol is not operationally up. This applies to IPCP, IPv6CP, OSINLCP, and MPLS.
  - lower layer down—Lower protocol layer is down. This applies to an interface only.

- max configure exceeded—Maximum number of configure requests was exceeded while negotiations were in progress. This means that there was no response from the peer, or the peer refused to negotiate. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.
- peer requested termination—Remote peer requested termination of the connection, which means that a terminate request was received while the session was in an open state. This applies to an interface, IPCP, IPv6CP, OSINLCP, and MPLS.

- Example 1—Provides detailed output for a particular IP interface

```
host1#show ppp interface atm 3/3.20 full
PPP interface ATM 3/3.20 is up
Interface alias is 'interface ezu19xuy'
Interface administrative status is open
Configured network protocol is IPCP
IPCP protocol configuration
   configured                    true
   administrative-status         open
   ip-address                    180.1.0.1
   dns-precedence                local
   wins-precedence               local
   ipcp-netmask-option           enabled
IPCP protocol status
   operational-status            up
IPCP negotiated options          local              peer
   ip-address                    180.1.0.1          195.0.1.13
   ip-address-mask               none               255.255.255.252
   primary-dns-address           none               192.168.10.10
   secondary-dns-address         none               none
   primary-wins-address          none               192.168.100.100
   secondary-wins-address        none               none
OSINLCP protocol configuration
   configured                    false
   administrative-status         open
OSINLCP protocol status
   operational-status            not present
   terminate-reason              not configured
MPLSNLCP protocol configuration
   configured                    false
   administrative-status         open
MPLSNLCP protocol status
   operational-status            not present
   terminate-reason              not configured
Interface statistics             in                 out
   packets                       0                  0
   octets                        617                1008
   errors                        0                  0
   discards                      384723             0
LCP protocol configuration
   max-receive-unit              use lower layer
   authentication                chap/pap
   magic-number                  enabled
   magic-number-mismatch         ignore
   keepalive-timer               0 seconds
   restart-timer                 3 seconds
   max-terminate                 2
   max-configure                 10
   max-failure                   5
   passive-mode                  disabled
```

```
LCP protocol status
   link-status                    network
LCP negotiated options            local              peer
   max-receive-unit               9178               9178
   authentication                 chap               none
   magic-number                   0x667cdfaa         0x27012f05
   accm                           none               none
   pfc                            none               none
   acfc                           none               none
LCP protocol statistics
   in-keepalive-requests          0
   out-keepalive-requests         0
   in-keepalive-replies           0
   out-keepalive-replies          0
   keepalive-failures             0
Authentication configuration
   authenticate-retry             0
   authentication-router          ''
Authentication status
   grant                          true
   session-timeout                none
   inactivity-timeout             none
   accounting-timeout             none
   peer-ip-address                none
   peer-ip-address-mask           255.255.255.252
   peer-primary-dns-address       192.168.10.10
   peer-secondary-dns-address     none
   peer-primary-wins-address      none
   peer-secondary-wins-address    none
Authentication statistics
   up-time                        53 seconds
   in-octets                      72
   out-octets                     60
   in-packets                     0
   out-packets                    0
PAP protocol configuration
   request-timeout                20 seconds
CHAP protocol configuration
   name                           ''
   challenge-retry                10
   challenge-timeout              4 seconds
   minimum-challenge-length       16
   maximum-challenge-length       32
   minimum-rechallenge-timeout    0 seconds
   maximum-rechallenge-timeout    0 seconds
```

- Example 2—Provides detailed output for a particular IPv6 interface

```
host1#show ppp interface fastEthernet 12/0.1.1 full
PPP interface FastEthernet 12/0.1.1 is lowerDown
Interface administrative status is open
Configured network protocol is IPV6CP
IPCP protocol configuration
   configured                     false
   administrative-status          open
   ip-address                     0.0.0.0
   dns-precedence                 local
   wins-precedence                local
   ipcp-netmask-option            disabled
IPCP protocol status
   operational-status             not present
```

```
IPV6CP protocol configuration
   configured                    true
   administrative-status         open
   ipv6-interfaceId              90:1a00:140:4b39
IPV6CP protocol status
   operational-status            down
   terminate-reason              link down
OSINLCP protocol configuration
   configured                    false
   administrative-status         open
OSINLCP protocol status
   operational-status            not present
Interface statistics             in                  out
   packets                       0                   0
   octets                        1163                706
   errors                        0                   0
   discards                      153482              0
LCP protocol configuration
   max-receive-unit              use lower layer
   authentication                none
   magic-number                  enabled
   magic-number-mismatch         reject
   keepalive-timer               30 seconds
   restart-timer                 3 seconds
   max-terminate                 2
   max-configure                 10
   max-failure                   5
   passive-mode                  disabled
LCP protocol status
   link-status                   initial
LCP protocol statistics
   in-keepalive-requests         11
   out-keepalive-requests        11
   in-keepalive-replies          11
   out-keepalive-replies         11
   keepalive-failures            0
Authentication configuration
   authenticate-retry            0
   authentication-router         ''
   aaa-profile                   ''
Authentication status
   grant                         false
   terminate-reason              lower layer down
PAP protocol configuration
   request-timeout               20 seconds
CHAP protocol configuration
   name                          ''
   challenge-retry               10
   challenge-timeout             4 seconds
   minimum-challenge-length      16
   maximum-challenge-length      32
   minimum-rechallenge-timeout   0 seconds
   maximum-rechallenge-timeout   0 seconds
```

- Example 3—Displays a termination reason (administrative disable) when the operational status of the interface is down

```
host1#show ppp interface
PPP interface pos 0/1:1 is lowerDown
PPP interface pos 4/0:1 is lowerDown
PPP interface pos 12/1:1 is lowerDown
3 ppp interfaces found
PPP interface serial 0/0:1/1 is Up
PPP interface serial 0/0:1/2 is Down (administrative disable)
```

show ppp interface summary

- Use to display a summary of all the multilinked and nonmultilinked PPP interfaces configured on the router.
- Field descriptions
  - PPP Status—Nonmultilinked PPP interfaces
  - Configuration status—Indicates the configuration state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS
    - configured—Interface or protocol is configured
    - notConfigured—Interface or protocol is not configured
  - Administrative status—Indicates the administrative state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS
    - open—Interface or protocol is administratively enabled
    - closed—Interface or protocol is administratively disabled
  - Operational status (Interface)—Indicates the operational state of the PPP interface
    - up—Interface is operational
    - down—Interface is not operational because of a problem in the PPP layer
    - lowerDown—Interface is not operational because a lower layer in the protocol stack is down
    - notPresent—Interface is not operational because the hardware is unavailable
    - passive—Interface is waiting for the peer to send an LCP confReq message
    - tunnel—Interface is being redirected through a tunnel
  - Operational status (Ip, Ipv6, Osi. Mpls)—Indicates the operational state of the IPCP, IPv6CP, OSINLCP, or MPLS protocol
    - up—Protocol is operational
    - down—Protocol is not operational because of a problem in the PPP layer
    - notPresent—Protocol is not operational because it does not exist
    - noResources—Protocol is not operational because it does not exist due to a lack of resources
  - PPP Multilink Status—Multilinked PPP interfaces

- Example

```
host1#show ppp interface summary
PPP Status
Configuration status     configured notConfigured
   Interface             4000       n/a
   Ip                    4000       0
   Ipv6                  0          4000
   Osi                   0          4000
   Mpls                  0          4000
Administrative status    open       closed
   Interface             4000       0
   Ip                    4000       0
   Ipv6                  4000       0
   Osi                   4000       0
   Mpls                  4000       0
Operational status       up         down       notPresent noResources
   Interface             4000       0          0          n/a
   Ip                    4000       0          0          0
   Ipv6                  0          0          4000       0
   Osi                   0          0          4000       0
   Mpls                  0          0          4000       0
Operational status       lowerDown  passive    tunnel
   Interface             0          0          0

PPP Multilink Status
Configuration status     configured notConfigured
   Link Interface        8000       n/a
   Network Interface     2000       n/a
   Ip                    2000       0
   Ipv6                  0          2000
   Osi                   0          2000
   Mpls                  0          2000
Administrative status    open       closed
   Link Interface        8000       0
   Network Interface     2000       0
   Ip                    2000       0
   Ipv6                  2000       0
   Osi                   2000       0
   Mpls                  2000       0
Operational status       up         down       notPresent noResources
   Link Interface        8000       0          0          n/a
   Network Interface     2000       0          0          n/a
   Ip                    2000       0          0          0
   Ipv6                  0          0          2000       0
   Osi                   0          0          2000       0
   Mpls                  0          0          2000       0
Operational status       lowerDown  passive    tunnel
   Link Interface        0          0          0
   Network Interface     0          0          0
```

Troubleshooting

Use the **pppPacket** log to diagnose problems on your PPP interfaces. On dynamic PPP interfaces, you can use the **ppp log** command within the profile, as described in *Chapter 15, Configuring Dynamic Interfaces*.

log severity debug pppPacket

- Use to configure a trace log file for a PPP interface.
- Specify one of the following interface types and an *interface specifier.* For example, specify *slot/port:channel/subchannel* for a serial POS PPP interface.
  - serial—Serial interface
  - atm—ATM interface
  - pos—Packet over SONET interface
- You also configure logging to direct the output to a specific destination. For information, see *JUNOSe System Event Logging Reference Guide, Chapter 1, System Logging Overview*.
- Example

```
host1(config-if)#log severity debug pppPacket serial 0/0:1/1
DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface:0/0:1/11/0:1,
time: 0.00, tx lcp confReq, id = 226, length = 19, mru = 32759,
authentication = chap MD5,magicNumber = 0x5387f9a2

DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface:
0/0:1/11/0:1,
time: 0.01, rx lcp confReq, id = 156, length = 18, mru = 32759,
magicNumber = 0x2d8eac91, pfc, acfc

DEBUG 01/01/1970 00:16:58 pppPacket (1000001,*): interface:
0/0:1/11/0:1,
time: 0.01, tx lcp confAck, id = 156, length = 18, mru = 32759,
magicNumber = 0x2d8eac91, pfc, acfc
```

- Use the **no** version to return the severity changes to their default setting or to the systemwide setting.

ppp log

- Use to enable PPP packet or state machine logging on any dynamic interface that uses the profile being configured. Specify one of the following keywords:
  - **pppPacket**—Enables PPP packet logging
  - **pppStateMachine**—Enables PPP state machine logging
- Example

```
host1(config-profile)#ppp log pppPacket
```

NOTE: This command is equivalent to the **log severity debug pppPacket** and **log severity debug pppStateMachine** commands.

- Use the **no** version to disable packet or state machine logging.

Chapter 8

Configuring Multilink PPP

This chapter describes how to configure a Multilink Point-to-Point Protocol (PPP) interface on E-series routers.

This chapter contains the following sections:

Overview

Multilink PPP (MLPPP; also referred to as PPP Multilink, MLP, and MP) aggregates multiple physical links into a single logical bundle. More specifically, MLPPP bundles multiple link-layer channels into a single network-layer channel. Peers negotiate MLPPP during the initial phase of Link Control Protocol (LCP) option negotiation. Each router indicates that it is multilink capable by sending the multilink option as part of its initial LCP configuration request.

An MLPPP bundle can consist of multiple physical links of the same type—such as multiple asynchronous lines—or can consist of physical links of different types—such as leased synchronous lines and dial-up asynchronous lines.

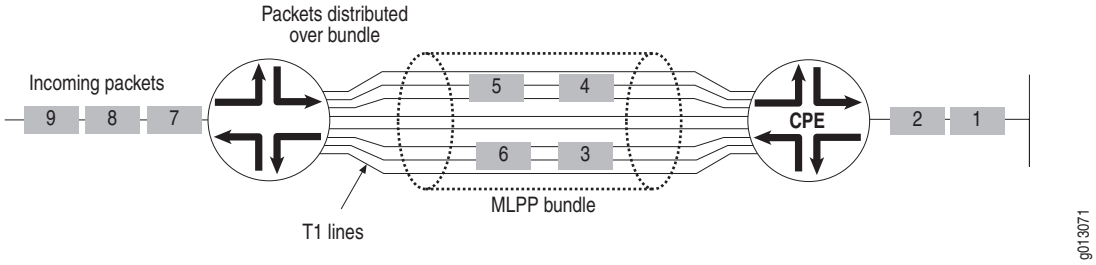
The router acts on MLPPP like another PPP Network Control Protocol (NCP). Packets received with an MLPPP header are subject to fragmentation, reassembly, and sequencing. Packets received without the MLPPP header cannot be sequenced and can be delivered only on a first-come, first-served basis.

Application

Some users need more bandwidth than a T1 or an E1 channel can provide, but cannot afford the expense or do not need the bandwidth of T3 or E3. Equal-cost multipath (ECMP) is one way to achieve the desired bandwidth. MLPPP is commonly used as an alternative to ECMP to deliver *NxT1* service. NxT1 service provides bandwidth greater than DS1 service without going up to the expense and infrastructure required for DS3 service. Cost-analysis of NxT1 versus DS3 service typically imposes a practical limit of 8xT1 service; that is, aggregation of no more than eight T1 or E1 connections into an MLPPP bundle.

The NxT1 implementation of MLPPP logically aggregates up to eight T1 or E1 connections into a single virtual connection, or bundle, to a given customer site, as shown in Figure 31.

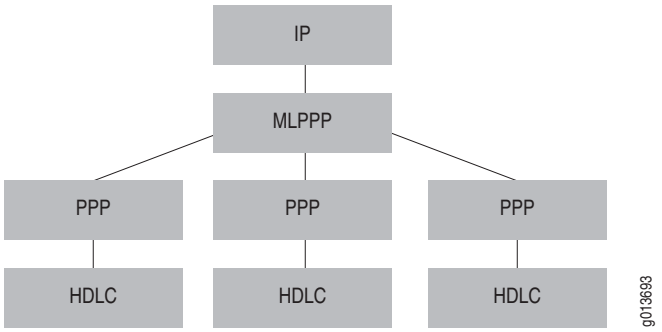
Figure 31: MLPPP Aggregation of T1 Lines into a Single Bundle



Because MLPPP aggregates multiple link-layer channels onto a single network-layer IP interface, protocol layering within the router is different than for non-multilink PPP.

Figure 32 illustrates interface stacking with MLPPP.

Figure 32: Structure of MLPPP



MLPPP LCP Extensions

Multilink PPP adds the following LCP negotiation options:

- Multilink maximum received reconstructed unit (MRRU) option—The MRRU option has two functions. First, it informs the other end of the link the maximum size of the PPP packet payload that the router can receive. Second, it informs the other end that the router supports MLPPP. When you enable multilink on your router, the router includes the MRRU option in LCP negotiation with the value set to the maximum received unit (MRU) value for PPP. If the remote system rejects this option, the local system determines that the remote system does not support multilink PPP and it terminates the link without negotiation.

NOTE: The router does not bring up a link if the MRU value received from a peer device differs from the MRRU value received from the peer.

- Short sequence number (SSN) header format option (not currently supported)—The SSN option indicates that the transmitting router wants to use a short sequence number (12 bits) in the MLPPP header rather than a long sequence number (24 bits). The router currently supports only long sequence numbers.

- Endpoint discriminator option—The endpoint discriminator option identifies the router transmitting the packet. If the receiving router determines that packets on another link have the same endpoint discriminator option, this link must be joined to that bundle. If the receiving router determines that no packets on other links have the same option, the receiving router must create a new bundle from this link.

  The endpoint discriminator is generated internally; you cannot configure it. The endpoint discriminator option is the same for all links on one end of the bundle; at the other end, all links also share a common endpoint discriminator. The two endpoint discriminators are different if the MLPPP bundle is set up between two E-series routers.

MLPPP Link Selection

By default, E-series routers use a round-robin algorithm to select the link on which to transmit data on an MLPPP interface. The round-robin link selection method applies to both best-effort packets, such as data, and non-best-effort (high-priority) packets, such as voice and video. Best-effort packets are encapsulated with an MLPPP header that contains a sequence number, whereas non-best-effort packets are encapsulated with a PPP header that does *not* contain a sequence number.

The member links in an MLPPP bundle can experience different queuing delays due to the volume of traffic transmitted on the MLPPP interface. These delays can cause packets to arrive out of order at the remote router. The effect of such delays differs for best-effort packets and non-best effort packets, as follows:

- For best-effort packets that arrive out of order from the E-series router, the remote router can use the sequence number to reorder and forward the packets in the correct order, regardless of the order in which the packets were received.

- For non-best-effort packets that arrive out of order from the E-series router, the lack of a sequence number prevents the remote router from being able to determine the correct order in which to forward the packets. This can cause problems with applications that require high-priority voice and video traffic transmitted on MLPPP interfaces to be received in the same order transmitted by the peer applications.

To ensure that the E-series router maintains the proper packet order when transmitting non-best-effort traffic, you can use the **ppp hash-link-selection** command to enable use of a hash-based algorithm to select the link on which the router transmits high-priority packets on an MLPPP interface.

When you use hash-based link selection instead of the default round-robin link selection for non-best-effort traffic, the router uses the IP source address (SA) and IP destination address (DA) of the packet as a hash to select the MLPPP member link on which to transmit the packet. Specifically, the router uses the hash algorithm to bind the transmission of all traffic between this IP SA and IP DA to the same member link in the MLPPP bundle.

If the member link selected to transmit high-priority packets becomes inoperable or is removed from the MLPPP bundle, the router must select a different link on which to transmit the packets. As a result, packets transmitted on this new link might sometimes arrive at the remote destination before the traffic sent on the previously selected member link.

You can configure hash-based MLPPP link selection in any of the following ways:

- To configure hash-based link selection for a individual MLPPP member link, issue the **ppp hash-link-selection** command from Interface Configuration mode or Subinterface Configuration mode in the context of the individual link interface. For more information, see *Configuring Static MLPPP* on page 266.

- To configure hash-based link selection for all current member links in an MLPPP bundle, issue the **ppp hash-link-selection** command from Interface Configuration mode in the context of the MLPPP bundle. Doing this has the same effect as issuing the **ppp hash-link-selection** command separately for each member link in the bundle. For more information, see *Contextual Command Differences* on page 267.

- To configure hash-based link selection for all dynamic MLPPP link interfaces created by a profile, issue the **ppp hash-link-selection** command from Profile Configuration mode. For more information, see *Configuring Dynamic MLPPP* on page 275.

For a detailed description and examples of using the **ppp hash-link-selection** command, see **ppp hash-link-selection** on page 271.

Platform Considerations

You can configure MLPPP interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support MLPPP interfaces on ERX-14xx models, ERX-7xx 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 MLPPP.

For information about the modules that support MLPPP interfaces 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 MLPPP.

Interface Specifiers

Some of the configuration task examples in this chapter use the *slot*/*port* format to specify the physical interface on which you want to configure MLPPP. However, the interface specifier format that you use depends on the type of physical interface on which you want to configure MLPPP and on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port* format. For example, the following command specifies an ATM interface on slot 5, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 5/1
```

For E120 and E320 routers, use the *slot*/*adapter*/*port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a tunnel-server port on slot 3, adapter 0, port 0 of an E320 router.

```
host1(config)#tunnel-server 3/0/0
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about the MLPPP protocol and MLPPP fragmentation and reassembly, consult the following resources:

- RFC 1661—The Point-to-Point Protocol (PPP) (July 1994)
- RFC 1990—The PPP Multilink Protocol (MP) (August 1996)
- RFC 2233—The Interfaces Group MIB using SMIv2 (November 1997)

Supported MLPPP Features

The router currently supports both the static configuration of the links participating in a multilink bundle and the dynamic creation of MLPPP bundles over L2TP (only on the LNS) when the LNS detects multilink LCP option negotiation in LCP proxy data.

The following MLPPP features are available for both static and dynamic MLPPP:

- Logical aggregation of up to eight links in a bundle
- Long sequence numbers
- Authentication for interfaces with MLPPP encapsulation or for MLPPP bundles
- Monotonically increasing sequence numbers

  All packets distributed across the member links have monotonically increasing sequence numbers. This feature enables the remote system on the customer premises to perform resequencing (if the system is configured to do so).

- Round-robin packet distribution or hash-based packet distribution

  By default, E-series routers use a round-robin algorithm to handle packet distribution across the member links in a bundle for both best-effort traffic and non-best-effort traffic. The round-robin approach is used even when the member links have different line rates.

As an alternative to round-robin packet distribution for non-best-effort traffic, you can enable use of a hash-based algorithm for distribution of non-best-effort (high-priority) packets, such as voice or video. Using a hash-based packet distribution mechanism instead of the default round-robin packet distribution mechanism for non-best-effort traffic ensures that the router maintains the proper packet order when transmitting high-priority packets. For details, see *MLPPP Link Selection* on page 257.

- Forwarding of multilink traffic to L2TP tunnels

  E-series routers support dynamic MLPPP over L2TP configurations (on the L2TP network server, or LNS).

- Fragmentation and reassembly

  For details, see *Configuring MLPPP Fragmentation and Reassembly* on page 276.

- Packet resequencing for best-effort traffic, for non-best-effort traffic, and when MLPPP reassembly is enabled

  For details on how the router supports packet resequencing for best-effort traffic and non-best-effort traffic, see *MLPPP Link Selection* on page 257.

  For details on enabling MLPPP reassembly, see *Configuring MLPPP Fragmentation and Reassembly* on page 276.

You can configure bundles as follows:

- On a COCX-F3 line module and its corresponding I/O modules, you can configure:
  - Up to 8 member links from different ports in the same bundle, with the following restriction for MLPPP reassembly:
    - For a COCX-F3 line module with either a 12-port E3-12 FRAME I/O module or a 12-port CT3/T3 12 I/O module, the restriction is based on the ports on which member links in the same bundle are configured.

      A 12-port E3-12 FRAME I/O module and a 12-port CT3/T3 12 I/O module each contain 12 ports numbered 0 through 11. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same port; on ports 0, 1, and 2; on ports 3, 4, and 5; on ports 6, 7, and 8; or on ports 9, 10, and 11. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span ports in different bundles; for example, on ports 0, 1, and 4.

      When MLPPP reassembly is disabled, this restriction is not in effect; that is, member links can span ports in different bundles.

  - Up to 12 bundles

- On a cOCx/STMx line module and its corresponding I/O module, you can configure:
  - Member links from different OC3/STM1 ports in the same bundle, with the following restrictions for MLPPP reassembly:
    - For a cOCx/STMx line module with a 4-port cOC3/STM1 I/O module, the restriction is based on the ports on which member links in the same bundle are configured.

      A 4-port cOC3/STM1 I/O module contains four ports numbered 0 through 3. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same port, on ports 0 and 1, or on ports 2 and 3. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span ports in different bundles; for example, on ports 1 and 2.

      When MLPPP reassembly is disabled, this restriction is not in effect; that is, member links can span ports in different bundles.

    - For a cOCx/STMx line module with a 1-port cOC12/STM4 I/O module, the restriction is based on the STM1 (OC3) paths on which member links in the same bundle are configured.

      A 1-port cOC12/STM4 I/O module has four logical paths numbered 1 through 4. When MLPPP reassembly is enabled, you can configure a bundle with member links on the same path, on paths 1 and 2, or on paths 3 and 4. However, the router *cannot* properly reassemble fragments if you configure a bundle with member links that span paths in different bundles; that is, on paths 2 and 3.

      When MLPPP reassembly is disabled, this restriction is not in effect; for example, member links can span paths in different bundles.

  - Any combination of bundles that does not exceed the 336 available T1 channels (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)
  - Any combination of bundles that does not exceed the 252 available E1 channels (for example, 252 single-link T1 bundles, 34 eight-link bundles, or 33 eight-link bundles and 8 single-link bundles)
- On a CT3/T3-F0 line module with a CT3/T3 12 I/O module, you can configure:
  - Member links from different T3 ports in the same bundle
  - Any combination of bundles that does not exceed the 336 available T1 channels (for example, 336 single-link T1 bundles, 42 eight-link bundles, or 41 eight-link bundles and 8 single-link bundles)

- On an ES2-S1 Service IOA, you can configure:
  - Up to 16,000 member links per line module, not to exceed a total of 12,000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 4000 four-link bundles, and 2000 eight-link bundles)
- On an OCx/STMx ATM line module and its corresponding line modules, you can configure:
  - Up to 8000 member links per line module, not to exceed a total of 8000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 2000 four-link bundles, and 1000 eight-link bundles)
- On a Service line module (SM), you can configure:
  - Up to 16,000 member links per line module, not to exceed a total of 12,000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 4000 four-link bundles, and 2000 eight-link bundles)
- On a shared tunnel-server port configured on a GE-2 or GE-HDE line module and corresponding line modules, you can configure:
  - Up to 8000 member links per line module, not to exceed a total of 8000 MLPPP bundles per chassis
  - Any combination of bundles that does not exceed either of these maximums (for example, 4000 single-link bundles, 4000 two-link bundles, 2000 four-link bundles, and 1000 eight-link bundles)
- On a ES2-S1 GE-4 IOA that pairs with an ES2 4G LM on E120 routers and E320 routers, you can configure:
  - MLPPP bundles with one or more links per bundle for dynamic MLPPP-over-PPPoE-over-Ethernet configurations.
  - MLPPP bundles with only one link per bundle when configuring static MLPPP-over-PPPoE-over-Ethernet. When you create multilink bundles in a static MLPPP-over-PPPoE-over-Ethernet configuration, PPPoE is unable to direct the PPPoE Active Discovery Initiation (PADI) packets received from the MLPPP bundle links on the client to the appropriate (matching) links in the MLPPP bundle on the server. As a result, the connections between bundle links become crossed, and the bundle does not come up as expected. Creating MLPPP bundles with only a single link for this configuration ensures a one-to-one correspondence between a PPPoE subscriber and its associated link, and guarantees that the MLPPP bundle comes up properly.

- MLPPP bundles with only a single link per bundle are *not* required for static MLPPP-over-PPPoE-over-Ethernet with VLAN configurations if all of the links in a bundle have the same VLAN ID that is unique across all MLPPP bundles configured on the line module.

- On all E-series ATM module combinations that support MLPPP, you can configure:

  - MLPPP bundles with one or more links per bundle for dynamic MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-ATM 1483 subinterface configurations.

  - MLPPP bundles with only one link per bundle when configuring static MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-an ATM 1483 subinterface. In this configuration, you can stack multiple PPPoE subinterfaces over a single PPPoE major interface.

  - Typically when you create ATM PVCs on an ATM module, there is a one-to-one correspondence between a PPPoE subscriber and the ATM PVC with which the subscriber is associated. However, in configurations with multiple PPPoE subinterfaces stacked over a single PPPoE major interface, crossed MLPPP bundle link connections can occur, as is the case with the ES2-S1 GE-4 IOA, and the bundle does not come up as expected. Creating MLPPP bundles with only a single link for this configuration ensures a one-to-one correspondence between a PPPoE subscriber and its associated link, and guarantees that the MLPPP bundle comes up properly.

  - MLPPP bundles with only a single link per bundle are *not* required for static MLPPP-over-multiple PPPoE subinterfaces-over-one PPPoE major interface-over-ATM 1483 subinterface configurations if all PPPoE subinterfaces stacked over the same PPPoE major interface belong to the same bundle.

NOTE: For information about the modules that support MLPPP on ERX-14xx models, ERX-7xx models, and the ERX-310 router, see *ERX Module Guide, Appendix A, Module Protocol Support*. For information about the modules that support MLPPP on the E120 router and the E320 router, see *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

Unsupported MLPPP Features

The router does not support the following MLPPP features:

- Short sequence numbers
- Resequencing out-of-order packets in the absence of fragmentation

  Given the location in the network where the router resides, the *N*xT1 links to a customer site represent one of many places across the IP network where packets might be received out of order. For example, if the router has multiple uplinks to a core router, packets might be received out of order across these links.

  You can lose packets if you transmit layer 2 traffic on an MPLS LSP that passes over an MLPPP link bundle.

  Packets are passed along to the next protocol layer in the order that they are processed. Packet resequencing is therefore performed at the end station rather than the aggregation router. IP datagrams can be resequenced by the end station using the IP identification field.

  Layer 2 packets such as Ethernet/MPLS and ATM-AAL5/MPLS have no sequence number information and are sent in the order received. The packets are dropped if their out-of-order condition is detected by a downstream device.

  Frame Relay/MPLS packets do have a native sequence number in the header and are rejected at the end of the LSP if the MLPPP sequence number order is violated.

  To ensure that the router maintains the proper packet order when transmitting high-priority (non-best-effort) packets such as voice and video, you can use the **ppp hash-link-selection** command to enable use of a hash-based algorithm to select the link on which the router transmits high-priority packets on an MLPPP interface. For details, see *MLPPP Link Selection* on page 257.

Before You Configure Static MLPPP

Before you begin configuring static MLPPP, you must configure the physical line interfaces to be aggregated by MLPPP. See the following chapters:

- *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 4, Configuring Channelized OCx/STMx Interfaces*

The procedures described in *Configuring Static MLPPP* on page 266 assume that a physical line interface has been configured.

Configuring Static MLPPP

Static MLPPP configuration consists of two general tasks, each with several subtasks.

To configure static MLPPP:

1. Create the member links to be aggregated into a multilink bundle.

   a. From Global Configuration mode, specify the individual interface on which you want to configure MLPPP.

      host1(config)#**interface serial 2/0:1/1**

   b. Specify MLPPP as the encapsulation method on the interface.

      host1(config-if)#**encapsulation mlppp**

   c. (Optional) Specify the keepalive timeout value for the member link interface.

      host1(config-if)#**ppp keepalive 50**

   d. (Optional) Specify the authentication method for the member link interface.

      host1(config-if)#**ppp authentication pap chap**

   e. (Optional) Enable hash-based link selection instead of the default round-robin link selection for the member link interface.

      host1(config-if)#**ppp hash-link-selection**

2. Add member links to a multilink bundle.

   a. Define the MLPPP bundle.

      host1(config)#**interface mlppp group1**

   b. Add each member link.

      host1(config-if)#**member-interface serial 2/0:1/1**

   c. Assign an IP address to the MLPPP bundle.

      host1(config-if)#**ip address 10.10.100.1 255.255.255.0**

   d. (Optional) Specify the keepalive timeout value for the MLPPP network interface (the entire MLPPP bundle).

      host1(config-if)#**ppp keepalive 50**

e. (Optional) Specify the authentication method for the MLPPP network interface (the entire MLPPP bundle).

```
host1(config-if)#ppp authentication pap chap
```

f. (Optional) Enable hash-based link selection instead of the default round-robin link selection for the MLPPP network interface (the entire MLPPP bundle).

```
host1(config-if)#ppp hash-link-selection
```

Configuration Example

The following commands configure three T1 lines and aggregate them into a multilink bundle named group1.

```
host1(config)#interface serial 2/0:1/1
host1(config-if)#encapsulation mlppp
host1(config-if)#exit
host1(config)#interface serial 2/0:2/1
host1(config-if)#encapsulation mlppp
host1(config-if)#exit
host1(config)#interface serial 2/0:3/1
host1(config-if)#encapsulation mlppp
host1(config-if)#ppp keepalive 50
host1(config-if)#exit
host1(config)#interface mlppp group1
host1(config-if)#member-interface serial 2/0:1/1
host1(config-if)#member-interface serial 2/0:2/1
host1(config-if)#member-interface serial 2/0:3/1
host1(config-if)#ppp authentication pap chap
host1(config-if)#ppp hash-link-selection
host1(config-if)#ip address 10.10.100.1 255.255.255.0
```

Contextual Command Differences

The MLPPP configuration commands have different effects depending on the interface context. If you issue an MLPPP configuration command in the context of an individual interface, the command affects only the MLPPP link interface associated with that individual interface.

For example, the following commands disable negotiation of the local magic number only for serial interface 2/0:1/1.

```
host1(config-if)#member-interface serial 2/0:1/1
host1(config-if)#encapsulation mlppp
host1(config-if)#ppp magic-number disable
```

If you issue an MLPPP configuration command in the context of an MLPPP bundle—the MLPPP network interface—the command affects all the member links of the bundle. This feature prevents you from having to issue MLPPP configuration commands for each member link interface.

For example, the following commands disable negotiation of the local magic number for the entire bundle, *group1*.

```
host1(config)#interface mlppp group1
host1(config-if)#member-interface serial 2/0:1/1
host1(config-if)#ip address 10.10.100.1 255.255.255.0
host1(config-if)#ppp magic-number disable
```

Any member links added to the bundle after issuing an MLPPP configuration command are not affected by the command. For example, if you add serial interface 2/0:4/1 to the *group1* bundle after you issue the **ppp magic-number disable** command, negotiation of the local magic number for this link and any member links subsequently added to the bundle is not disabled.

Configuring Authentication

Perform the following optional tasks to configure authentication on interfaces with MLPPP encapsulation or MLPPP bundles.

- Specify one or more PPP authentication types.
- Modify the length of the CHAP challenge.
- Specify the maximum number of retries.

NOTE: The JUNOSe software's PPP application accepts null usernames during PAP and CHAP authentication. When the PPP application receives an authentication request that includes a null username, PPP passes the request to AAA. To take advantage of this feature, configure your authentication server to support the use of null usernames.

ppp authentication

- Use to require authentication from the PPP peer.
- To specify the name of a virtual router (VR) to be used as the authentication VR context, use the **virtual-router** keyword. Keep the following points in mind when you use the **ppp authentication virtual-router** command:
  - When you specify a VR in the **ppp authentication** command, AAA does not query the domain map for the assigned VR context. Instead, AAA uses the VR specified in the **ppp authentication** command as the authentication VR context and issues the authentication request to the authentication server in the assigned VR context.
  - If you specify the default VR as the authentication VR context, AAA loosely binds the user to the default VR. This means that RADIUS *can override* the default VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies the default VR, AAA returns either the default VR or the VR specified by RADIUS.

  - If you specify a VR other than the default VR as the authentication VR, AAA tightly binds the user to the specified VR. This means that RADIUS *cannot override* the specified VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies a nondefault VR, AAA returns the specified VR.
- The router supports the MD5 authentication algorithm for CHAP authentication.
- Example 1—Specify PAP or CHAP as the primary authentication protocol, and the other authentication protocol as the alternative. For example, the following command specifies **pap** as the primary authentication protocol and **chap** as the alternate.

  host1(config-if)#**ppp authentication pap chap**

  The router requests the use of PAP as the authentication protocol (because it appears first in the command line). If the peer refuses to use PAP, the router requests the CHAP protocol. If the peer refuses to negotiate authentication, the router terminates the PPP session.

- Example 2—Specify a virtual router for the authentication virtual router context. This command is available in static configurations and in profiles.

  host1(config-if)#**ppp authentication virtual-router boston pap chap**

- Use the **no** version to specify that the router does not require authentication.

ppp chap-challenge-length

- Use to modify the length of the CHAP challenge by specifying the allowable minimum length and maximum length.

CAUTION: Do *not* use the **ppp chap-challenge-length** command; increasing the minimum length (from the default 16 bytes) or decreasing the maximum length (from the default 32 bytes) reduces the security of your router.

- Specify the minimum and maximum lengths in bytes in the range 8–63.
- The maximum length must be greater than or equal to the minimum length.
- Example

  host1(config-if)#**ppp chap-challenge-length 24 28**

- Use the **no** version to restore the default minimum (16 bytes) and default maximum (32 bytes).

ppp max-bad-auth

- Use to specify the maximum number of authentication retries the router allows before terminating a PPP session
- This value applies to PAP and CHAP authentication.
- The range is 0–7. The default is 0, which indicates that no retries are allowed.

- Example

  host1(config-if)#**ppp max-bad-auth 3**

- Use the **no** version to return the number of retries to the default, 0.

Configuring Other PPP Attributes

The available **ppp** command options are the same for interfaces whether they are configured with PPP or MLPPP.

encapsulation mlppp

- Use to configure MLPPP as the encapsulation method on an individual interface.
- Use this command only within the context of an individual interface. Issuing this command creates an MLPPP link interface, also referred to as an MLPPP bundle member.
- Example

  host1(config)#**interface serial 2/0:1/1**
  host1(config-if)#**encapsulation mlppp**

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

interface mlppp

- Use to create an MLPPP network interface, also known as the MLPPP bundle.
- Example

  host1(config-if)#**interface mlppp group2**

- Use the **no** version to delete the MLPPP bundle. You must first delete the IP interface, followed by deleting the bundle members (link interfaces); then you can delete the MLPPP bundle.

NOTE: RADIUS supports the inclusion of the MLPPP Bundle Name VSA [26-62] in Access-Request, Acct-Start, Acct-Stop, and Interim-Acct messages. For more information, see *JUNOSe Broadband Access Configuration Guide, Chapter 3, Configuring RADIUS Attributes*.

member-interface

- Use to add an MLPPP link interface—also known as an MLPPP bundle member—to an MLPPP bundle.
- Example

  host1(config-if)#**member-interface serial 2/0:1/1**

- Use the **no** version to remove the specified interface from the MLPPP bundle.

ppp hash-link-selection

- Use to enable use of a hash-based algorithm to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on an MLPPP interface.
- Hash-based MLPPP link selection is available only for non-best-effort traffic. For best-effort traffic, the router uses a round-robin algorithm for link selection.
- Using hash-based link selection instead of the default round-robin link selection for non-best-effort traffic ensures that the router maintains the proper packet order when transmitting high-priority packets.
- When you configure hash-based link selection, the router uses the IP source address and IP destination address of the packet as a hash to select the MLPPP member link on which to transmit the packet.
- You can configure hash-based MLPPP link selection in any of the following ways:
  - To configure hash-based link selection for an individual MLPPP member link interface, issue the **ppp hash-link-selection** command from Interface Configuration mode or Subinterface Configuration mode in the context of the link interface. (See Example 1.)
  - To configure hash-based link selection for all current member links in an MLPPP bundle, issue the **ppp hash-link-selection** command from Interface Configuration mode in the context of the MLPPP bundle. (See Example 2.)
  - To configure hash-based link selection for all dynamic MLPPP link interfaces created by a profile, issue the **ppp hash-link-selection** command from Profile Configuration mode. (See Example 3.)
- Example 1—The following commands configure hash-based MLPPP link selection for an individual MLPPP member link interface.

```
host1(config)#interface atm 2/0
host1(config-if)#interface atm 2/0.2
host1(config-subif)#atm pvc 42 0 42 aal5snap
host1(config-subif)#encapsulation mlppp
host1(config-subif)#ppp hash-link-selection
```

- Example 2—The following commands configure hash-based MLPPP link selection for all current member links in the MLPPP bundle (group1). Doing this has the same effect as issuing the **ppp hash-link-selection** command separately for each member link in the bundle.

```
host1(config)#interface mlppp group1
host1(config-if)#ppp hash-link-selection
```

- Example 3—The following commands configure hash-based MLPPP link selection for all dynamic MLPPP interfaces created by the profile named dynamicMlppp.

```
host1(config)#profile dynamicMlppp
host1(config-profile)#ppp multilink enable
host1(config-profile)#ppp hash-link-selection
```

- Use the **no** version to restore the default round-robin algorithm for MLPPP link selection.

ppp keepalive

- Use to specify the keepalive timeout value in the range 10–64800 seconds. If issued in the context of an individual interface, the command affects only that interface. If issued in the context of an MLPPP bundle, the command affects all MLPPP link interfaces that are member links of that bundle.
- When the keepalive timer expires, the interface always sends an LCP echo request, regardless of whether the peer is silent.
- When the keepalive interval is 30 seconds (the default), a failed link is detected between 90 and 120 seconds after failure.
- Use **ppp keepalive** without a value to restore the default, 30 seconds.
- Example

  host1(config-if)#**ppp keepalive 50**

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

ppp log

- Use to enable PPP packet or state machine logging on any dynamic interface that uses the profile being configured. Specify one of the following keywords:
  - **pppPacket**—Enables PPP packet logging
  - **pppStateMachine**—Enables PPP state machine logging
- Example

  host1(config-profile)#**ppp log pppPacket**

NOTE: This command is equivalent to the **log severity debug pppPacket** and **log severity debug pppStateMachine** commands.

- Use the **no** version to disable packet or state machine logging.

ppp magic-number disable

- Use to disable negotiation of the local magic number. If issued in the context of an individual interface, the command affects only that interface. If issued in the context of an MLPPP bundle, the command affects all MLPPP link interfaces that are member links of that bundle.
- Issuing this command prevents the router from detecting loopback configurations.
- Example

  host1(config-if)#**ppp magic-number disable**

- Use the **no** version to restore negotiation of the local magic number.

ppp magic-number ignore-mismatch

- Use to cause the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.
- For more information about using this command, see *Validation of LCP Peer Magic Number* in *Chapter 7, Configuring Point-to-Point Protocol*.
- To verify configuration of LCP peer magic number validation on the router, use the **show ppp interface mlppp** command. For information, see **show ppp interface mlppp** on page 288.
- Example

  host1(config-if)#**ppp magic-number ignore-mismatch**

- Use the **no** version to restore the default behavior, in which the router terminates the PPP connection if it detects an LCP peer magic number mismatch.

ppp mru

- Use to control the negotiation of the maximum receive unit (MRU).
- Specify the number of bytes, in the range 64–65535.
- We recommend you coordinate this value with the network administrator on the other end of the line.
- If the value configured for the PPP MRU is greater than the value of the lower-layer MRU minus the PPP header length, the router logs a warning message and uses the lesser of the configured MRU value or the lower-layer MRU value minus the PPP header length to negotiate the local MRU.
- If the value configured for the PPP MRU conflicts with a similar value configured for another protocol, such as the MTU value for PPPoE, the router uses the lesser of the two values.
- If you issue the command in the context of an encapsulated MLPPP interface, it affects only that interface. If you issue the command in the context of an MLPPP bundle, it affects all member links within that bundle.
- Example

  host1(config-if)#**ppp mru 576**

- Use the **no** version to restore the default value, which causes PPP to use the lower-layer MRU minus the PPP header length as the MRU value.

ppp passive-mode

- Use to force a static or dynamic PPP interface into passive mode, for a period of one second, before LCP negotiation begins. This delay enables slow clients to start up and initiate the LCP negotiation.
- Example

  host1(config-if)#**ppp passive-mode**

- Use the **no** version to disable passive mode.

ppp peer

- Use to resolve conflicts when the system and the PPP peer system have primary and secondary DNS and WINS addresses configured with different values.
- By default, the DNS and WINS addresses configured on the system take precedence.
- Use the **ppp peer dns** or the **ppp peer wins** commands to configure the PPP peer system as the one that takes precedence. This command has no effect unless both systems have the address configured and the address is in conflict. If the PPP peer system has the address and the system does not, the peer always supplies the address regardless of how you have configured the PPP peer.
- Example

  host1(config-profile)#**ppp peer dns**

- Use the **no ppp peer dns** or the **no ppp peer wins** commands when you want the system to take precedence during setup negotiations between the system and the remote PC client. If the IP addresses passed to the system by the remote PC client differ from the ones you have configured on your system, the system returns the values that you configured as the correct values to the remote PC client.

ppp shutdown

- Use to terminate an MLPPP session.
- If you use the **ip** or **osi** keyword, disables the Internet Protocol Control Protocol (IPCP) or OSI Network Layer Control Protocol (OSINLCP) service for the MLPPP network interface (MLPPP bundle). Issue only in the context of a network interface.
- If no keywords are issued, issuing this command has the following effect:
  - If issued in the context of an individual interface, the command affects only that interface. The **ip** and **osi** keywords are not functional in this context.
  - If issued in the context of an MLPPP bundle, the command affects all MLPPP link interfaces that are member links of that bundle. The **ip** and **osi** keywords are functional only in this context.
- The **ppp shutdown** command administratively disables the interface.
- Example

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

- If you issue the **ppp shutdown** command in the context of an MLPPP bundle, you cannot bring up an individual member link by subsequently issuing the **no ppp shutdown** command in the context of that member. You can bring up only the entire bundle; to do so, you must issue the **no ppp shutdown** command in the context of the bundle. If you add new member links while a bundle is shut down, those new members are also in the shut-down state until the entire bundle is brought up.
- Use the **no** version to restart a disabled session.

Configuring Dynamic MLPPP

You can define a profile to dynamically create MLPPP bundles over L2TP on the LNS. The profile consists of commands to define the bundle attributes, just as you would for static configuration. For more information about profiles for dynamic interfaces, see *Chapter 15, Configuring Dynamic Interfaces*.

To configure a profile for dynamic MLPPP:

1. Create a profile by assigning it a name.

   host1(config)#**profile dynmlppp**

2. Enable creation of dynamic MLPPP interfaces.

   host1(config-profile)#**ppp multilink enable**

3. Specify a virtual router to which dynamic IP interfaces created using this profile will be assigned.

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

4. Specify an IP loopback interface with which dynamic IP interfaces created using this profile will be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

5. (Optional) Set other desired PPP characteristics by using the **ppp** commands described in *Configuring Authentication* on page 268 and *Configuring Other PPP Attributes* on page 270.

ppp multilink enable

- Use in a profile to enable the creation of dynamic MLPPP interfaces.
- Example

  host1(config-profile)#**ppp multilink enable**

- Use the **no** version to cause the LNS to reject any incoming requests to create dynamic MLPPP interfaces.

profile

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

  host1(config)#**profile dynmlppp1**

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

Configuring MLPPP Fragmentation and Reassembly

You can configure MLPPP fragmentation and reassembly on a static link interface before adding the link to a bundle, or in a profile assigned to a dynamic MLPPP interface. You can also configure fragmentation and reassembly for all current member links in an MLPPP bundle.

Overview

E-series routers support fragmentation and reassembly as part of their MLPPP implementation. *Fragmentation* is the process by which a large packet is broken up into multiple smaller fragments for simultaneous transmission across multiple links of an MLPPP bundle. *Reassembly* is the process by which the destination router reassembles the fragments into the original packets.

Application

You can use MLPPP fragmentation and reassembly to reduce transmission latency. You can also use the feature to implement a packet-prioritization scheme that allows smaller, delay-sensitive packets (such as high-priority voice packets) to be interleaved with or race ahead of larger, delay-insensitive packets (such as low-priority data packets) when they are transmitted in the network.

Supported Configurations

Table 13 lists the static and dynamic MLPPP configurations on E-series routers that support fragmentation and reassembly.

Table 13: Supported Configurations for MLPPP Fragmentation and Reassembly

Static MLPPP Configurations	Dynamic MLPPP Configurations
Static MLPPP over ATM 1483 subinterfaces	Dynamic MLPPP over ATM 1483 subinterfaces
Static MLPPP over PPPoE over ATM 1483 subinterfaces	Dynamic MLPPP over PPPoE over ATM 1483 subinterfaces
Static MLPPP over serial (HDLC) interfaces	Dynamic MLPPP over serial (HDLC) interfaces
–	Dynamic MLPPP over L2TP (on the L2TP network server)

Module Requirements

For a list of the line modules and corresponding I/O modules that support MLPPP fragmentation and reassembly on ERX-7xx models, ERX-14xx models, and the ERX-310 router, see *ERX Module Guide*, *Appendix A*, *Module Protocol Support*.

For a list of the line modules and corresponding IOAs that support MLPPP fragmentation and reassembly on the E120 router and the E320 router, see *E120 and E320 Module Guide, Appendix A, IOA Protocol Support*.

Link Configuration Parameters

The parameters for MLPPP fragmentation and reassembly are configured on a per-link basis for each link interface (also known as a member link) in an MLPPP bundle.

By default, fragmentation and reassembly are disabled for MLPPP links. You can enable or disable fragmentation and reassembly for an individual link, or for all member links in a bundle, by using the **ppp fragmentation** and **ppp reassembly** commands. However, you must configure the same fragmentation setting and the same reassembly setting—enabled or disabled—for all member links in a bundle.

When you use the **ppp fragmentation** command to enable fragmentation on a link, you can optionally specify the maximum fragment size to be used on the link interface. When you use the **ppp reassembly** command to enable reassembly on a link, you can optionally specify the administrative multilink maximum received reconstructed unit (MRRU) value for the link.

Bundle Validation and Configuration Guidelines

When you configure MLPPP, the router validates that each link interface attempting to join a statically or dynamically created bundle has Link Control Protocol (LCP) parameters that are compatible with the other member links already in the bundle. This validation includes examining the parameters configured for fragmentation and reassembly on a particular link interface and verifying that these parameters are compatible with the other member links in the bundle.

To ensure that the bundle validation succeeds, make sure you observe the following configuration guidelines for MLPPP fragmentation and reassembly.

Guidelines for MLPPP Fragmentation

Use the following guidelines when you configure MLPPP fragmentation on a link interface:

- Configure the same fragmentation setting—enabled or disabled—for all member links in a bundle.
- When fragmentation is enabled, configure the same fragment size for all member links in a bundle.
- Make sure a link's fragment size does not exceed its maximum transmission unit (MTU) size.
- 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.

Guidelines for MLPPP Reassembly

Use the following guidelines when you configure MLPPP reassembly on a link interface:

- Configure the same reassembly setting—enabled or disabled—for all member links in a bundle.
- Make sure a link's administrative MRRU is greater than or equal to the local maximum receive unit (MRU) negotiated both on that link and on other member links in the bundle.

- The local MRRU negotiated on a link must be the same as the local MRRU negotiated on the other member links in the bundle.
- The peer MRRU negotiated on a link must be the same as the peer MRRU negotiated on the other member links in the bundle.
- When reassembly is enabled, member links belonging to the same bundle can have different local MRU values.
- When reassembly is disabled, member links belonging to the same bundle must negotiate the same local MRU value.

Bundle Validation Failure

If an MLPPP link interface fails bundle validation because one or more of the preceding configuration guidelines are not met, the router's actions differ depending on whether you are using a static MLPPP configuration or a dynamic MLPPP configuration, as follows:

- For static MLPPP configurations, the router permits the failed link to join the bundle, but forces the link into a down state.
- For dynamic MLPPP configurations, the router prohibits the failed link from joining the bundle, and subsequently tears down the link.

Recovering from Bundle Validation Failure

To recover from a bundle validation failure, you must reconfigure the link interface (for static MLPPP configurations) or reconfigure the profile (for dynamic MLPPP configurations) according to the guidelines described in *Bundle Validation and Configuration Guidelines* on page 277.

Configuring Fragmentation and Reassembly for Static MLPPP

To configure fragmentation and reassembly on a static MLPPP link interface:

1. From Global Configuration mode, specify the individual link interface on which you want to configure fragmentation and reassembly.

   host1(config)#**interface serial 4/0:1/1/1/1/1**

2. Specify MLPPP as the encapsulation method on the link interface.

   host1(config-if)#**encapsulation mlppp**

3. Enable fragmentation on the link interface, and optionally specify the maximum allowable fragment size to use.

   host1(config-if)#**ppp fragmentation 128**

NOTE: You can specify the maximum fragment size for a link only when you use the **ppp fragmentation** command to enable fragmentation on that link. You cannot specify the maximum fragment size for a link when fragmentation is disabled.

4. Enable reassembly on the link interface, and optionally specify the administrative MRRU value to use.

   host1(config-if)#**ppp reassembly 1590**

NOTE: You can specify the administrative MRRU value for a link only when you use the **ppp reassembly** command to enable reassembly on that link. You cannot specify the administrative MRRU for a link when reassembly is disabled.

5. Exit Interface Configuration mode.

   host1(config-if)#**exit**

6. Repeat Steps 1 through 5 for each additional link interface on which you want to configure fragmentation and reassembly. For example:

   host1(config)#**interface serial 4/0:1/1/1/1/2**
   host1(config-if)#**encapsulation mlppp**
   host1(config-if)#**ppp fragmentation 128**
   host1(config-if)#**ppp reassembly 1590**
   host1(config-if)#**exit**

7. Define the MLPPP bundle.

   host1(config)#**interface mlppp group1**

8. Add each member link to the bundle.

   host1(config-if)#**member-interface serial 4/0:1/1/1/1/1**
   host1(config-if)#**member-interface serial 4/0:1/1/1/1/2**

9. Assign an IP address to the MLPPP bundle.

   host1(config-if)#**ip address 10.10.100.1 255.255.255.0**

Static MLPPP over ATM 1483 Example

The following example configures MLPPP fragmentation and reassembly for two member links in an MLPPP bundle over an ATM 1483 subinterface.

```
host1(config)#interface atm 2/0
host1(config-if)#interface atm 2/0.2
host1(config-subif)#atm pvc 42 0 42 aal5snap
host1(config-subif)#encapsulation mlppp
host1(config-subif)#ppp fragmentation
host1(config-subif)#ppp reassembly 1400
host1(config-subif)#ppp authentication pap chap
host1(config-subif)#exit
host1(config)#interface atm 2/0.3
host1(config-subif)#atm pvc 43 0 43 aal5snap
host1(config-subif)#encapsulation mlppp
host1(config-subif)#ppp fragmentation
host1(config-subif)#ppp reassembly 1600
host1(config-subif)#ppp authentication pap chap
host1(config-subif)#exit
```

```
host1(config)#interface mlppp client1
host1(config-if)#member-interface atm 2/0.2
host1(config-if)#member-interface atm 2/0.3
host1(config-if)#ip address 10.10.200.1 255.255.255.0
```

Configuring Fragmentation and Reassembly for Dynamic MLPPP

To configure fragmentation and reassembly for dynamic MLPPP, you must create a profile that includes commands to define the link and bundle attributes, just as you do for a static MLPPP configuration.

For more information, see:

- *Chapter 15, Configuring Dynamic Interfaces*
- *JUNOSe Broadband Access Configuration Guide, Chapter 12, Configuring an L2TP LAC*
- *JUNOSe Broadband Access Configuration Guide, Chapter 13, Configuring an L2TP LNS*

To define a profile that configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface:

1. From Global Configuration mode, create a profile by assigning it a name, and access Profile Configuration mode.

   ```
   host1(config)#profile dynmlppp1
   host1(config-profile)#
   ```

2. Enable the creation of dynamic MLPPP interfaces.

   ```
   host1(config-profile)#ppp multilink enable
   ```

3. Enable fragmentation on the link interface, and optionally specify the maximum allowable fragment size to use.

   ```
   host1(config-profile)#ppp fragmentation 128
   ```

NOTE: You can specify the maximum fragment size for a link only when you use the **ppp fragmentation** command to enable fragmentation on that link. You cannot specify the maximum fragment size for a link when fragmentation is disabled.

4. Enable reassembly on the link interface, and optionally specify the administrative MRRU value to use.

   ```
   host1(config-profile)#ppp reassembly 1800
   ```

NOTE: You can specify the administrative MRRU value for a link only when you use the **ppp reassembly** command to enable reassembly on that link. You cannot specify the administrative MRRU for a link when reassembly is disabled.

5. (Optional) Specify a virtual router to which dynamic IP interfaces created with this profile will be assigned.

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

6. (Optional) Specify an IP loopback interface with which dynamic IP interfaces created with this profile will be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

7. (Optional) Set other PPP characteristics as needed by using the **ppp** commands described in *Chapter 8, Configuring Multilink PPP*.

Dynamic MLPPP over PPPoE Example

The following example configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface over dynamic PPPoE over an ATM 1483 subinterface.

```
host1(config)#profile dynmlppp2
host1(config-profile)#ppp multilink enable
host1(config-profile)#ppp fragmentation 128
host1(config-profile)#ppp reassembly 1800
host1(config-profile)#ip virtual-router westford
host1(config-profile)#ip unnumbered loopback 1
host1(config-profile)#pppoe sessions 9
host1(config-profile)#ppp authentication chap
host1(config-profile)#exit
host1(config)#interface atm 4/0
host1(config-if)#interface atm 4/0.1
host1(config-subif)#atm pvc 52 0 52 aal5autoconfig 0 0 0
host1(config-subif)#profile pppoe dynmlppp2
host1(config-subif)#auto-configure pppoe
```

Dynamic MLPPP over L2TP Example

The following example configures MLPPP fragmentation and reassembly for a dynamic MLPPP interface over L2TP over a Gigabit Ethernet interface.

```
host1(config)#ip router-id 193.1.1.1
host1(config)#interface loopback 0
host1(config-if)#ip address 193.1.1.1 255.255.255.0
host1(config-if)#interface gigabitEthernet 1/1
host1(config-if)#ip unnumbered loopback 0
host1(config-if)#exit
host1(config)#ip route 193.1.1.2 255.255.255.255 gigabitEthernet 1/1
host1(config)#profile l2tp-profile
host1(config-profile)#ip virtual-router default
host1(config-profile)#ip unnumbered loopback 0
host1(config-profile)#ip access-routes
host1(config-profile)#ppp authentication pap
host1(config-profile)#ppp keepalive
host1(config-profile)#ppp multilink enable
host1(config-profile)#ppp mru 1590
host1(config-profile)#ppp reassembly 1590
host1(config-profile)#ppp fragmentation 128
host1(config-profile)#pppoe session 8000
host1(config-profile)#exit
```

```
host1(config)#l2tp destination profile lac ip address 193.1.1.2
host1(config-l2tp-dest-profile)#remote host xxx.com
host1(config-l2tp-dest-profile-host)#enable proxy authenticate
host1(config-l2tp-dest-profile-host)#tunnel password welcome
host1(config-l2tp-dest-profile-host)#profile l2tp-profile
```

encapsulation mlppp

- Use to configure MLPPP as the encapsulation method on an individual interface.
- Use this command only within the context of an individual interface. Issuing this command creates an MLPPP link interface, which can be configured as a member of an MLPPP bundle.
- Example

  ```
  host1(config)#interface serial 2/0:1/1
  host1(config-if)#encapsulation mlppp
  ```

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

interface mlppp

- Use to create an MLPPP network interface, also known as an MLPPP bundle.
- Example

  ```
  host1(config-if)#interface mlppp group2
  ```

- Use the **no** version to delete the MLPPP bundle. To delete an MLPPP bundle you must first delete the IP interface, then delete the bundle members (link interfaces), and finally delete the MLPPP bundle itself.

member-interface

- Use to add an MLPPP link interface—also known as an MLPPP bundle member—to an MLPPP bundle.
- Example

  ```
  host1(config-if)#member-interface serial 2/0:1/1
  ```

- Use the **no** version to remove the specified interface from the MLPPP bundle.

ppp fragmentation

- Use to enable fragmentation on an MLPPP link interface.
- If fragmentation is enabled on the link, you can optionally specify the maximum fragment size to be used on that link, in the range 128–65535 octets.
- A link's maximum fragment size cannot exceed the MTU size on that link.
- We recommend that all member links in an MLPPP bundle be assigned the same fragment size.

- Do not configure both MLPPP fragmentation 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)#**ppp fragmentation 128**

- Use the **no** version to disable fragmentation on the link and restore the default fragment size, which is the link's MTU.

ppp multilink enable

- Use in a profile to enable the creation of dynamic MLPPP interfaces.
- Example

  host1(config-profile)#**ppp multilink enable**

- Use the **no** version to reject any incoming requests to create dynamic MLPPP interfaces.

ppp reassembly

- Use to enable reassembly on an MLPPP link interface.
- If reassembly is enabled on the link, you can optionally specify the administrative MRRU for the link, in the range 64–65535 octets. The administrative MRRU is the maximum allowable size of the PPP packet payload that the router can receive.
- A link's MRRU must be greater than or equal to the local MRU on that link.
- We recommend that all member links in an MLPPP bundle be assigned the same reassembly setting: enabled or disabled.
- Example

  host1(config-if)#**ppp reassembly 1590**

- Use the **no** version to disable reassembly on the link and restore the default value, which is the local MRU on the link.

profile

- Use to create a profile for a dynamic interface.
- You specify a profile name of up to 80 characters.
- Example

  host1(config)#**profile dynmlppp1**

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

Configuring Fragmentation and Reassembly for MLPPP Bundles

If you issue the **ppp fragmentation** command or the **ppp reassembly** command in the context of an MLPPP bundle, the command affects all the current member links in the bundle. This enables you to issue a single command for the entire bundle instead of having to issue individual commands for each member link in the bundle.

For example, the following commands configure MLPPP fragmentation and reassembly for all member links in the bundle group1.

```
host1(config)#interface mlppp group1
host1(config-if)#ppp fragmentation 128
host1(config-if)#ppp reassembly 1590
host1(config-if)#exit
host1(config)#
```

Any member links added to the bundle after you issue an MLPPP configuration command in the bundle context are not affected by the command. For example, if you add a member link to the group1 bundle after you issue the **ppp fragmentation** or **ppp reassembly** command, MLPPP fragmentation and reassembly for this link and any member links subsequently added to the bundle is not enabled.

Monitoring MLPPP

Use the commands in this section to display information about MLPPP interfaces.

You can set a statistics baseline for MLPPP serial (member link) or bundle (multilink) interfaces using the **baseline ppp** command. Use the **delta** keyword with the **show** commands described below to display statistics with the baseline values subtracted.

After you configure multilink PPP, you can use the **show ppp interface** commands to display configuration and statistics information about MLPPP and MLPPP fragmentation and reassembly.

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. For details, see *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

baseline ppp interface

- Use to set a statistics baseline for PPP interfaces—including MLPPP interfaces, either individual serial (member link) interfaces or multilink (bundle) interfaces.
- Use only the **serial** or **mlppp** keywords.
- For serial interfaces, specify the interface location in the format *slot*/*port*:*channel*/*subchannel* for CT3 modules.

- For MLPPP interfaces, specify the interface location as the name of the MLPPP bundle.
- 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.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline was not set, the following message is displayed instead:

  ```
  No baseline has been set
  ```

- Use the optional **delta** keyword with MLPPP **show** commands to specify that baselined statistics are to be shown.
- Example

  host1#**baseline ppp interface serial 2/0:1/1**

- There is no **no** version.

Sample Display Without Baseline

The following command displays PPP interface (including MLPPP interface) statistics *without* baselining:

```
host1#show ppp interface statistics

PPP interface serial 2/0:4/1 is up
No baseline has been set
Interface statistics              in                out
   packets                        0                 0
   octets                         572               684
   errors                         0                 0
   discards                       0                 0
PPP interface serial 2/0:5/1 is up
No baseline has been set
Interface statistics              in                out
   packets                        0                 0
   octets                         572               684
   errors                         0                 0
   discards                       0                 0
PPP interface serial 2/1:4/1 is up
No baseline has been set
Interface statistics              in                out
   packets                        0                 0
   octets                         572               684
   errors                         0                 0
   discards                       0                 0
PPP interface serial 2/1:5/1 is up
No baseline has been set
Interface statistics              in                out
   packets                        0                 0
   octets                         572               684
   errors                         0                 0
   discards                       0                 0
4 ppp interfaces found
```

```
PPP interface mlppp group1 is up
PPP multilink member-interface serial 2/0:1/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        608                 716
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/0:2/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        608                 716
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/0:3/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        596                 704
   errors                        0                   0
   discards                      0                   0

PPP interface mlppp group2 is up
PPP multilink member-interface serial 2/1:1/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        628                 740
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/1:2/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        628                 740
   errors                        0                   0
   discards                      0                   0
PPP multilink member-interface serial 2/1:3/1 is up
No baseline has been set
Interface statistics             in                  out
   packets                       0                   0
   octets                        616                 728
   errors                        0                   0
   discards                      0                   0
2 mlppp interfaces found
```

Sample Display with Baseline

The following command displays PPP interface (including MLPPP interface) statistics *with* baselining:

```
host1#show ppp interface statistics delta

PPP interface serial 2/0:4/1 is up
Time since last baseline 00:00:35
Interface statistics             in                  out
   packets                       0                   0
   octets                        75                  82
   errors                        0                   0
   discards                      0                   0
PPP interface serial 2/0:5/1 is up
Time since last baseline 00:00:37
```

```
Interface statistics              in                  out
   packets                        0                   0
   octets                         87                  90
   errors                         0                   0
   discards                       0                   0
PPP interface serial 2/1:4/1 is up
Time since last baseline 00:00:39
Interface statistics              in                  out
   packets                        0                   0
   octets                         101                 112
   errors                         0                   0
   discards                       0                   0
PPP interface serial 2/1:5/1 is up
Time since last baseline 00:00:43
Interface statistics              in                  out
   packets                        0                   0
   octets                         94                  99
   errors                         0                   0
   discards                       0                   0
4 ppp interfaces found
PPP interface mlppp group1 is up
PPP multilink member-interface serial 2/0:1/1 is up
Time since last baseline 00:00:17
Interface statistics              in                  out
   packets                        0                   0
   octets                         28                  26
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface serial 2/0:2/1 is up
Time since last baseline 00:10:22

Interface statistics              in                  out
   packets                        0                   0
   octets                         102                 104
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface serial 2/0:3/1 is up
Time since last baseline 00:00:19
Interface statistics              in                  out
   packets                        0                   0
   octets                         112                 126
   errors                         0                   0
   discards                       0                   0

PPP interface mlppp group2 is up
PPP multilink member-interface serial 2/1:1/1 is up
Time since last baseline 00:00:23
Interface statistics              in                  out
   packets                        0                   0
   octets                         125                 132
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface serial 2/1:2/1 is up
Time since last baseline 00:00:25
Interface statistics              in                  out
   packets                        0                   0
   octets                         135                 138
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface serial 2/1:3/1 is up
Time since last baseline 00:00:30
```

```
Interface statistics             in                out
   packets                       0                 0
   octets                        125               132
   errors                        0                 0
   discards                      0                 0
2 mlppp interfaces found
```

show ppp interface mlppp

- Use to display information about MLPPP interfaces.
- You can display a great variety of information with this complex command. See the **show ppp interface** command in *Chapter 7, Configuring Point-to-Point Protocol*, for more detailed information about the display options.
- Use the **show ppp interface** command to display information about all PPP interfaces, including MLPPP interfaces.
- Field descriptions
  - PPP interface mlppp—Name and administrative status (up or down) for an MLPPP bundle
  - PPP multilink member-interface—Interface type, interface specifier, and administrative status (up or down) for an MLPPP member link
  - Network interface administrative status—Indicates whether the interface for the MLPPP bundle is administratively enabled (open), meaning that the **no ppp shutdown** command is operational, or administratively disabled (closed), meaning that the **ppp shutdown** command is operational
  - Link interface administrative status—Indicates whether the interface for the member link is administratively enabled (open), meaning that the **no ppp shutdown** command is operational, or administratively disabled (closed), meaning that the **ppp shutdown** command is operational
  - Configured network protocol—Network protocol configured on the interface
  - Fragmentation and reassembly configuration:
    - Link interface fragmentation—Indicates whether MLPPP fragmentation is enabled or disabled on the link interface
    - Link interface fragment size—MLPPP fragment size, in octets, currently in use on the link interface
    - Link interface reassembly—Indicates whether MLPPP reassembly is enabled or disabled on the link interface
    - Link interface administrative MRRU—Administrative MRRU value, in octets, currently in use on the link interface
  - Baseline status—Indicates whether a statistics baseline has been set
  - Interface statistics:
    - packets—Number of packets received (in) and sent (out) on the interface
    - octets—Number of octets received (in) and sent (out) on the interface

- errors—Number of errors received (in) and sent (out) on the interface
- discards—Number of packets discarded on receipt (in) or discarded before they were transmitted (out)

NOTE: For the LCP, IPCP, and OSINLCP negotiated options, the command displays a value of "none" if the option was not negotiated.

- LCP protocol configuration:
  - max-receive-unit—Controls negotiation of the local MRU option; value can be one of the following:
    - use lower layer—MRU of the layer below PPP defines the MRU to be negotiated
    - disabled—MRU option is not to be negotiated
    - a numeric value—MRU value to be negotiated
  - authentication—Controls negotiation of the local authentication option; value can be one of the following:
    - none—Do not negotiate
    - chap—Negotiate CHAP
    - pap—Negotiate PAP
    - chap/pap—Negotiate CHAP, and if it is rejected, negotiate PAP
    - pap/chap—Negotiate PAP, and if it is rejected, negotiate CHAP
  - magic-number—Controls whether the local magic number is negotiated: enabled (negotiate), or disabled (do not negotiate)
  - magic-number-mismatch—Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
  - keepalive-timer—Rate of LCP echo requests, in seconds
  - restart-timer—Retry frequency during LCP, IPCP, and OSINLCP negotiations, in seconds
  - max-terminate—Maximum number of terminate requests
  - max-configure—Maximum number of configure requests
  - max-failure—Maximum number of configure NAKs
- LCP protocol status:
  - link-status—Indicates the overall status of LCP negotiations, including the following states: initial (idle), starting (ready to negotiate), authenticate (authenticating), and network (LCP is up)

- LCP negotiated options:
  - max-receive-unit—Negotiated maximum receive unit, in octets, for the local and remote (peer) side of the link
  - max-receive-reconstructed-unit—Negotiated maximum receive reconstructed unit, in octets, for the local and remote (peer) side of the link
  - authentication—Negotiated authentication method (none, pap, or chap) for the local and remote (peer) side of the link
  - magic-number—Negotiated magic number for the local and remote (peer) side of the link
  - pfc—Negotiated pfc (none or enabled) for the local and remote (peer) side of the link
  - acfc—Negotiated acfc (none or enabled) for the local and remote (peer) side of the link

NOTE: The command displays a value of "none" for any negotiated option parameters if the option was not negotiated.

- LCP Endpoint Discriminator options:
  - local discriminator class—Endpoint discriminator type, format, and address space for the local system
  - local endpoint discriminator—Endpoint discriminator value for the local system within the specified class
  - peer discriminator class—Endpoint discriminator type, format, and address space for the remote system
  - peer endpoint discriminator—Endpoint discriminator value for the remote system within the specified class
- LCP protocol statistics:
  - in-keepalive-requests—Number of received keepalive requests (LCP Echo Request) for the life of the interface (since either system boot or interface creation, whichever is later)
  - out-keepalive-requests—Number of transmitted keepalive requests for the life of interface
  - in-keepalive-replies—Number of received keepalive replies for the life of the interface
  - out-keepalive-replies—Number of transmitted keepalive replies for the life of the interface
  - keepalive-failures—Number of keepalive failures reported on the interface
- IPCP protocol configuration:
  - configured—IPCP is configured on this interface (true or false)
  - administrative-status—IPCP administrative status (open or closed)
  - ip-address—Address to be used for negotiation of local IP address option

- dns-precedence—Used to resolve conflicts during DNS address negotiation
- local—Local side takes precedence, and the **no ppp peer dns** command is operative
- peer—Remote side takes precedence, and the **ppp peer dns** command is operative
- wins-precedence—Used to resolve conflicts during WINS address negotiation
- local—Local side takes precedence, and the **no ppp peer wins** command is operative
- peer—Remote side takes precedence, and the **ppp peer wins** command is operative

- IPCP protocol status:
  - operational-status—IPCP operational status (up, down, not present, or not present no resources)
- IPCP negotiated options:
  - ip-address—Negotiated IP address for the local and remote (peer) side of the link
  - primary-dns-address—Negotiated primary DNS address for the local and remote (peer) side of the link
  - secondary-dns-address—Negotiated secondary DNS address for the local and remote (peer) side of the link
  - primary-wins-address—Negotiated primary WINS address for the local and remote (peer) side of the link
  - secondary-wins-address—Negotiated secondary WINS address for the local and remote (peer) side of the link

NOTE: The command displays a value of "none" for any negotiated option parameters if the option was not negotiated.

- OSINLCP protocol configuration:
  - configured—OSINLCP is configured on this interface (true or false)
  - administrative-status—OSINLCP administrative status (open or closed)
- OSINLCP protocol status:
  - operational-status—OSINLCP operational status (up, down, not present, or not present no resources)
  - terminate-reason—Reason for termination of OSINLCP service
- OSINLCP negotiated options
  - npdu-alignment—Negotiated NPDU alignment for the local and remote (peer) side of the link

- Example 1—Displays information about the MLPPP member links configured in bundle group1

```
host1#show ppp interface mlppp group1 members
PPP interface mlppp group1 is up
 PPP multilink member-interface serial 2/0:1/1 is up
 PPP multilink member-interface serial 2/0:2/1 is up
 PPP multilink member-interface serial 2/0:3/1 is up
```

- Example 2—Displays information about all MLPPP member links configured for all bundles

```
host1#show ppp interface mlppp members
PPP interface mlppp group1 is up
 PPP multilink member-interface serial 2/0:1/1 is up
 PPP multilink member-interface serial 2/0:2/1 is up
 PPP multilink member-interface serial 2/0:3/1 is up
PPP interface mlppp group2 is up
 PPP multilink member-interface serial 2/1:1/1 is up
 PPP multilink member-interface serial 2/1:2/1 is up
 PPP multilink member-interface serial 2/1:3/1 is up
PPP interface mlppp group3
 No member-interfaces found
```

- Example 3—Displays information about all MLPPP encapsulated links, regardless of whether the links are members of an MLPPP bundle

```
host1#show ppp interface mlppp links
PPP multilink interface serial 2/0:1/1 is up
PPP multilink interface serial 2/0:2/1 is up
PPP multilink interface serial 2/0:3/1 is up
PPP multilink interface serial 2/1:1/1 is up
PPP multilink interface serial 2/1:2/1 is up
PPP multilink interface serial 2/1:3/1 is up
```

- Example 4—Displays configuration information about MLPPP member links configured in bundle group1

```
host1#show ppp interface mlppp group1 config
PPP interface mlppp group1 is up
Network interface administrative status is open
Configured network protocol is IPCP
PPP multilink member-interface ATM 10/0.10 is up
Link interface administrative status is open
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.11 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.12 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
PPP multilink member-interface ATM 10/0.13 is down (lower layer down)
Link interface administrative status is closed
Link interface fragmentation is enabled
```

```
Link interface fragment size is 128
Link interface reassembly is enabled
Link interface administrative MRRU is 2000
1 mlppp interfaces found
```

- Example 5—Displays statistics about all configured MLPPP member links configured in bundle group1

```
host1#show ppp interface mlppp group1 statistics
PPP interface mlppp group1 is up
PPP multilink member-interface ATM 10/0.10 is up
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         170                 690
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface ATM 10/0.11 is down (lower layer down)
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         50                  0
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface ATM 10/0.12 is down (lower layer down)
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         50                  0
   errors                         0                   0
   discards                       0                   0
PPP multilink member-interface ATM 10/0.13 is down (lower layer down)
No baseline has been set
Interface statistics              in                  out
   packets                        0                   0
   octets                         50                  0
   errors                         0                   0
   discards                       0                   0
1 mlppp interfaces found
```

- Example 6—Displays status information about the specified MLPPP bundle

```
host1#show ppp interface mlppp group1 status
PPP interface mlppp group1 is up
1 mlppp interfaces found
```

- Example 7—Shows complete configuration, statistics, and status information about the specified MLPPP bundle

```
host1#show ppp interface mlppp group1 full
PPP interface mlppp group1 is up
Network interface administrative status is open
Configured network protocol is IPCP
IPCP protocol configuration
   configured                     true
   administrative-status          open
   ip-address                     1.2.3.4
   dns-precedence                 local
   wins-precedence                local
IPCP protocol status
   operational-status             up
```

```
IPCP negotiated options           local              peer
   ip-address                     1.2.3.4            6.7.8.9
   primary-dns-address            none               none
   secondary-dns-address          none               none
   primary-wins-address           none               none
   secondary-wins-address         none               none
OSINLCP protocol configuration
   configured                     false
   administrative-status          open
OSINLCP protocol status
   operational-status             not present
   terminate-reason               not configured
PPP multilink member-interface serial 2/0:1/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics              in                 out
   packets                        0                  0
   octets                         1488               1972
   errors                         0                  0
   discards                       0                  0
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network

LCP negotiated options            local              peer
   max-receive-unit               1590               1590
   max-receive-reconstructed-unit 1590               1590
   authentication                 none               none
   magic-number                   0x6c079eb0         0x2c5a5798
   pfc                            none               none
   acfc                           none               none
LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x313939333130303038000001b000001
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x313939333130303038000001b000002
LCP protocol statistics
   in-keepalive-requests          70
   out-keepalive-requests         70
   in-keepalive-replies           70
   out-keepalive-replies          70
   keepalive-failures             0
PPP multilink member-interface serial 2/0:2/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics              in                 out
   packets                        0                  0
   octets                         1508               1996
   errors                         0                  0
   discards                       0                  0
```

```
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network
LCP negotiated options            local              peer
   max-receive-unit               1590               1590
   max-receive-reconstructed-unit 1590               1590
   authentication                 none               none
   magic-number                   0x7ada4a05         0x1bb178cd
   pfc                            none               none
   acfc                           none               none

LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x3139393331303030380000 1b000001
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x3139393331303030380000 1b000002
LCP protocol statistics
   in-keepalive-requests          71
   out-keepalive-requests         71
   in-keepalive-replies           71
   out-keepalive-replies          71
   keepalive-failures             0
PPP multilink member-interface serial 2/0:3/1 is up
Link interface administrative status is open
No baseline has been set
Interface statistics              in                 out
   packets                        0                  0
   octets                         1568               2068
   errors                         0                  0
   discards                       0                  0
LCP protocol configuration
   max-receive-unit               use lower layer
   authentication                 none
   magic-number                   enabled
   magic-number-mismatch          ignore
   keepalive-timer                30 seconds
   restart-timer                  3 seconds
   max-terminate                  2
   max-configure                  10
   max-failure                    5
LCP protocol status
   link-status                    network
LCP negotiated options            local              peer
   max-receive-unit               1590               1590
   max-receive-reconstructed-unit 1590               1590
   authentication                 none               none
   magic-number                   0x31cc52e0         0x32ebdec6
   pfc                            none               none
   acfc                           none               none
LCP Endpoint Discriminator options
   local discriminator class      Locally Assigned Address
   local endpoint discriminator   0x3139393331303030380000 1b000001
   peer discriminator class       Locally Assigned Address
   peer endpoint discriminator    0x3139393331303030380000 1b000002
```

```
LCP protocol statistics
   in-keepalive-requests          74
   out-keepalive-requests         74
   in-keepalive-replies           74
   out-keepalive-replies          74
   keepalive-failures             0
1 mlppp interfaces found
```

show ppp interface summary

- Use to display a summary of all the multilinked and nonmultilinked PPP interfaces configured on the router.
- Field descriptions
  - PPP Status—Non-multilinked PPP interfaces
  - Configuration status—Indicates the configuration state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS
    - configured—Interface or protocol is configured
    - notConfigured—Interface or protocol is not configured
  - Administrative status—Indicates the administrative state of the PPP interface, IPCP, IPv6CP, OSINLCP, or MPLS
    - open—Interface or protocol is administratively enabled
    - closed—Interface or protocol is administratively disabled
  - Operational status (Interface)—Indicates the operational state of the PPP interface
    - up—Interface is operational
    - down—Interface is not operational because of a problem in the PPP layer
    - lowerDown—Interface is not operational because a lower layer in the protocol stack is down
    - notPresent—Interface is not operational because the hardware is unavailable
    - passive—Interface is waiting for the peer to send an LCP confReq message
    - tunnel—Interface is being redirected through a tunnel
  - Operational status (Ip, Ipv6, Osi, Mpls)—Indicates the operational state of the IPCP, IPv6CP, OSINLCP, or MPLS protocol
    - up—Protocol is operational
    - down—Protocol is not operational because of a problem in the PPP layer
    - notPresent—Protocol is not operational because it does not exist
    - noResources—Protocol is not operational because it does not exist due to a lack of resources
  - PPP Multilink Status—Multilinked PPP interfaces

- Example

```
host1#show ppp interface summary
PPP Status
Configuration status     configured notConfigured
   Interface             4000       n/a
   Ip                    4000       0
   Ipv6                  0          4000
   Osi                   0          4000
   Mpls                  0          4000
Administrative status    open       closed
   Interface             4000       0
   Ip                    4000       0
   Ipv6                  4000       0
   Osi                   4000       0
   Mpls                  4000       0
Operational status       up         down       notPresent noResources
   Interface             4000       0          0          n/a
   Ip                    4000       0          0          0
   Ipv6                  0          0          4000       0
   Osi                   0          0          4000       0
   Mpls                  0          0          4000       0
Operational status       lowerDown  passive    tunnel
   Interface             0          0          0

PPP Multilink Status
Configuration status     configured notConfigured
   Link Interface        8000       n/a
   Network Interface     2000       n/a
   Ip                    2000       0
   Ipv6                  0          2000
   Osi                   0          2000
   Mpls                  0          2000
Administrative status    open       closed
   Link Interface        8000       0
   Network Interface     2000       0
   Ip                    2000       0
   Ipv6                  2000       0
   Osi                   2000       0
   Mpls                  2000       0
Operational status       up         down       notPresent noResources
   Link Interface        8000       0          0          n/a
   Network Interface     2000       0          0          n/a
   Ip                    2000       0          0          0
   Ipv6                  0          0          2000       0
   Osi                   0          0          2000       0
   Mpls                  0          0          2000       0
Operational status       lowerDown  passive    tunnel
   Link Interface        0          0          0
   Network Interface     0          0          0
```

Chapter 9
Configuring Packet over SONET

Use the procedures described in this chapter to configure packet over SONET (POS) on E-series routers.

This chapter contains the following sections:

Overview

Packet over SONET (Synchronous Optical Network)/SDH (Synchronous Digital Hierarchy) is the serial transmission of data over SONET frames through the use of a protocol such as Point-to-Point Protocol (PPP).

Packet over SONET/SDH is an ideal feature for networks that are built for providing Internet or IP data. It provides superior bandwidth utilization and efficiency compared with other transport methods. For expensive WAN links, packet over SONET can provide as much as 25 to 30 percent higher throughput than networks based on Asynchronous Transfer Mode (ATM). By transporting frames directly into the SONET/SDH payload, the overhead required in an ATM cell header for IP over ATM encapsulation is eliminated.

The router supports PPP, Cisco High-Level Data Link Control (HDLC), and Frame Relay over SONET/SDH.

POS Features

POS supports the following features:

- Payload scrambling
- Clock source configuration
- Maximum transmission unit (MTU) size configuration
- Maximum receive unit (MRU) size configuration
- POS framing
- Cyclic redundancy check (CRC) checking
- Loopback configuration

SONET/SDH

SONET is an ANSI standard for transmitting bits over fiber-optic cable. SDH is the international standard defined by the International Telecommunication Union (ITU). SONET/SDH is the physical infrastructure of choice for carrier ATM networks operating at speeds above 50 Mbps.

SONET/SDH allows carriers to build high-speed international links without requiring conversion from one transmission protocol to another (for example, T1 to T3 or T1 to E3 conversion).

SONET transmission speeds start at 51.84 Mbps and are referred to as OC1. SDH transmission speeds start at 155.52 Mbps and are referred to as STM1. All other speeds are multiples of these base numbers. Table 14 shows the speeds of the most common SONET/SDH implementations.

Table 14: Most Common SONET/SDH Implementations

SONET	SDH	Transmission Speed
OC1	—	51.84 Mbps
OC3	STM1	155.52 Mbps
OC12	STM4	622.08 Mbps
OC48	STM16	2.4 Gbps
OC96	STM32	4.876640 Gbps
OC192	STM64	9.953280 Gbps

Platform Considerations

You can configure POS interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support POS interfaces on ERX-14xx models, ERX-7xx 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 POS.

For information about the modules that support POS interfaces 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 POS.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify a POS interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies a POS interface on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface pos 0/1
```

For E120 and E320 routers, use the *slot*/*adapter*/*port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a POS interface on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface pos 5/0/0
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about POS interfaces, consult the following resources:

- RFC 1662—PPP in HDLC-like Framing (July 1994)
- RFC 2615—PPP over SONET/SDH (June 1999)
- RFC 2427—Multiprotocol Interconnect over Frame Relay (September 1998)

Before You Configure POS

Before you configure a POS interface, verify that you have correctly installed the required module. For information about installing modules in ERX-7xx models, ERX-14xx models, and ERX-310 routers, see *ERX Hardware Guide, Chapter 4, Installing Modules*. For information about installing modules in the E120 router and the E320 router, see *E120 and E320 Hardware Guide*, *Chapter 4*, *Installing Modules*. Then verify that no ATM interfaces are defined on the physical port.

Also have the following information available:

- Interfaces specifiers for the POS interfaces that you want to create

  For more information about specifying POS interfaces on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

- IP addresses and subnet mask assignments for IP interfaces

Configuration Tasks

To configure a POS interface:

1. Configure a physical interface.

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

2. (Optional) Assign a text description or an alias to the interface.

   host1(config-if)#**pos description austin01 pos interface**

3. Configure the encapsulation method.

   host1(config-if)#**encapsulation ppp**

4. (Optional) Configure the internal clock source.

   host1(config-if)#**clock source internal module**

5. (Optional) Set the size of the CRC.

   host1(config-if)#**crc 32**

6. (Optional) Set the time interval at which the router calculates bit and packet rate counters.

   host1(config-if)#**load-interval 90**

7. (Optional) Set the type of loopback mode.

   host1(config-if)#**loopback line**

8. (Optional) Set the MRU size.

   host1(config-if)#**mru 1000**

9. (Optional) Set the MTU size.

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

10. (Optional) Set the type of framing.

    host1(config-if)#**pos framing sdh**

11. Disable payload scrambling.

    host1(config-if)#**no pos scramble-atm**

12. (Optional) Disable an interface.

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

clock source

- Use to set the clock source.
- You can set **internal** or **line** clocking.
- Internal clocking has two options:
  - **module**—Uses internal clock from the line module
  - **chassis**—Uses the configured router clock
- Example

  host1(config-if)#**clock source internal module**

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

crc

- Use to set the number of bits used for CRC checking.
- CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data; 16 and 32 indicate the number of check digits per frame that are used to calculate the frame check sequence (FCS). Both the sender and receiver must use the same setting.
- Example

  host1(config-if)#**crc 32**

- Use the **no** version to restore the default value, 16.

encapsulation frame-relay ietf

- Use to specify Frame Relay as the encapsulation method for the interface.
- The router uses IETF format (RFC 2427 encapsulation).
- Example

  host1(config-if)#**encapsulation frame-relay ietf**

- Use the **no** version to remove the Frame Relay configuration from an interface.

encapsulation ppp

- Use to specify PPP as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation ppp**

- Use the **no** version to remove the PPP configuration from an interface.

interface pos

- Use to configure a POS interface.
- To specify a POS interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface
- To specify a POS interface for E120 and E320 routers, use the *slot*/*adapter*/*port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- For more information about modules that support POS interfaces, see *JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*.
- Examples

  host1(config-if)#**interface pos 0/1**
  host1(config-if)#**interface pos 5/0/0**

- Use the **no** version to remove the POS interface.

load-interval

- Use to set the time interval at which the router calculates bit and packet rate counters.
- You can choose a multiple of 30 seconds, in the range 30–300 seconds.
- Example

  host1(config-if)#**load-interval 90**

- Use the **no** version to restore the default value, 300.

loopback

- Use to specify the type of loopback for a POS interface.
  - **internal**—Connects the local transmitted signal to the local received signal.
  - **line**—Connects the received network signal directly to the transmit network signal. When configured in line loopback mode, the router never receives data from the network.

- Example

  host1(config-if)#**loopback line**

- Use the **no** version to clear the loopback.

mru

- Use to set the maximum allowable size of the MRU.
- Specify a value in the range 1–9996 bytes.
- Example

  host1(config-if)#**mru 1000**

- Use the **no** version to restore the default value, 4470.

mtu

- Use to set the maximum allowable size of the MTU.
- Specify a value in the range 1–9996 bytes.
- Example

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

- Use the **no** version to restore the default value, 4470.

pos description

- Use to assign a text description or an alias to a POS HDLC interface.
- You can use this command to help you identify the interface and keep track of interface connections.
- The description or alias can be a maximum of 80 characters.
- Use the **show interfaces pos** command to display the text description.
- Example

  host1(config-if)#**pos description austin01 pos interface**

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

pos framing

- Use to set the type of framing for a POS interface.
  - **sdh**—Uses SDH framing format
  - **sonet**—Uses SONET framing format (the default)
- Example

  host1(config-if)#**pos framing sdh**

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

pos scramble-atm

- Use to enable payload scrambling on a POS interface.
- Payload scrambling is enabled by default. When enabled, both sides of the connection must be using the scrambling algorithm. The router uses a 43rd-order synchronous scrambler to scramble the output data.
- Example

  host1(config-if)#**pos scramble-atm**

- Use the **no** version to disable scrambling on the POS interface.

shutdown

- Use to disable a POS interface.
- Example

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

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

Monitoring POS

Use the **show interfaces pos** command to display information about the POS interface. You can set a statistics baseline for POS interfaces using the **baseline interface pos** command.

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 *show Commands* in *JUNOSe System Basics Configuration Guide, Chapter 2, Command-Line Interface*, for details.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

baseline interface pos

- Use to set a statistics baseline for POS 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.
- Example

  host1#**baseline interface pos 8/0**

- There is no **no** version.

show interfaces pos

- Use to display the configuration, state, and statistics for a POS interface.
- To specify a POS interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface
- To specify a POS interface for E120 and E320 routers, use the *slot*/*adapter*/*port* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
- You can include the following keywords:
  - **delta**—Specifies that baselined statistics are to be shown
  - **brief**—Displays the operational status of all configured interfaces
- Field descriptions
  - POS interface status—State of the physical interface: up, down
  - Description—Text description or alias if configured for the interface
  - snmp trap link-status—SNMP trap status: disabled: up, down
  - Encapsulation—Layer 2 encapsulation display; options: ppp, frame-relay ietf, mlppp, mlframe-relay ietf, hdlc
  - SONET path operational status—State of the SONET path: up, down, lowerLayerDown
  - time since last status change—Last reported change to the SONET path operational status
  - SONET operational status—State of SONET operation: up, down, lowerLayerDown
  - time since last status change—Last reported change to the SONET operational status
  - loopback—Loopback status for the physical interface: enabled, disabled
  - timing source—Clocking source for the physical interface
  - framing type—Framing type for the physical interface
  - Crc type checking—Number of bits used for CRC checking: crc16, crc32, none
  - Hdlc mru—MRU size allowed on the interface
  - Hdlc mtu—MTU size allowed on the interface

- Hdlc interface speed—Line speed of the interface
- Hdlc scrambling—Status of payload scrambling on the interface: on, off
- 5 minute input rate—Data rates based on the traffic received in the last five minutes
- 5 minute output rate—Data rates based on the traffic sent in the last five minutes
- Packets received—Number of incoming packets received on this interface
- Bytes received—Number of incoming bytes received on this interface
- Errored packets received—Number of incoming errors received on this interface
- Packets sent—Number of outgoing packets transmitted on this interface
- Bytes sent—Number of outgoing bytes transmitted on this interface
- Errored packets sent—Number of outgoing errors on this interface

- Example

```
host1#show interfaces pos 8/0
Packet over SONET interface 8/0 is ifOperUp
Description: houston80 pos interface
snmp trap link-status = disabled
Encapsulation ppp
SONET path operational status: up
      time since last status change: 00:20:37
SONET operational status:      up
      time since last status change: 00:20:37
loopback not set
timing source is loop timing
framing type is SONET
Crc type checking - CRC32
Hdlc mru = 4470
Hdlc mtu = 4470
Hdlc interface speed = 155520000
Hdlc scrambling is off
5 minute input rate 24910848 bits/sec, 1023242 packets/sec
5 minute output rate 24905728 bits/sec, 1023233 packets/sec

Interface statistics
Packets received              1066995954
Bytes received                3836558195
Errored packets received      0
Packets sent                  1055275550
Bytes send                    3039550548
Errored packets sent          0
```

Chapter 10

Configuring Point-to-Point Protocol over Ethernet

This chapter describes how to configure the Point-to-Point Protocol (PPP) over Ethernet interfaces on E-series routers.

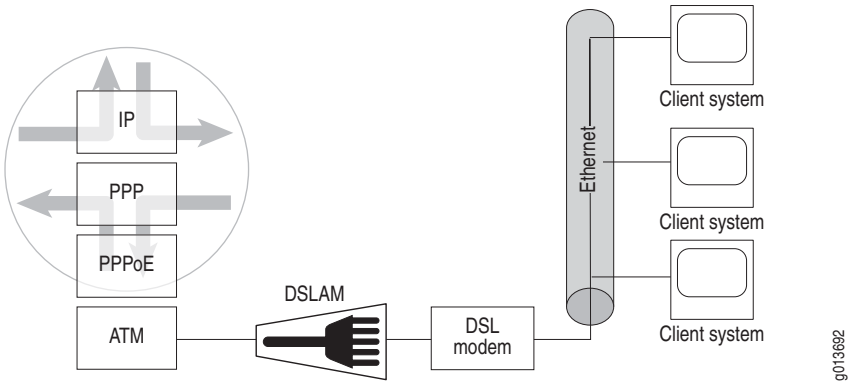
This chapter contains the following sections:

Overview

E-series routers use PPP over Ethernet (PPPoE) to enable multiple hosts to open PPP sessions to the router using one or more bridging modems. When service providers want to maintain the session abstraction associated with PPP, PPPoE is used with Broadband Remote Access Server (B-RAS) technologies that provide a bridged Ethernet topology. PPPoE can be configured over ATM or on Ethernet modules with or without VLANs.

Figure 33 shows how PPPoE allows the router to handle multiple PPP sessions originating on an Ethernet module to be multiplexed over one PVC on an ATM interface. PPP, as described in *Chapter 7, Configuring Point-to-Point Protocol*, runs above the PPPoE layer.

Figure 33: PPPoE over ATM



The router handles the server part of PPPoE session management and never initiates a setup of a PPPoE session. The router only responds to session requests that are sent to it by the remote PPP client. After the sessions are set up, the router demultiplexes the sessions based on session identifiers assigned to a specific connection.

PPPoE Stages

PPPoE has two distinct stages: Discovery and Session.

Discovery

PPPoE includes a Discovery protocol that allows each PPP session to learn the Ethernet address of the remote peer, as well as establish a unique session identifier. When a host wants to initiate a PPPoE session, it must first perform Discovery to identify the Ethernet MAC address of the peer and establish a PPPoE session ID.

Although PPP defines a peer-to-peer relationship, Discovery is inherently a client-server relationship. In the Discovery process, a host acting as a client discovers a remote access concentrator (AC), which acts as the server.

Based on the network topology, there may be more than one remote AC with whom the host can communicate. The Discovery stage allows the host to discover all remote ACs and then select the one to which it wants to connect.

In summary, the Discovery stage consists of the following four steps:

1. The host (PPPoE client) broadcasts a PPPoE Active Discovery Initiation (PADI) packet to all remote ACs in the network.

2. One or more remote ACs respond to the PADI packet by sending a PPPoE Active Discovery Offer (PADO) packet, indicating that they can serve the client request. The PADO packet includes the name of the AC from which it was sent.

3. The host sends a unicast PPPoE Active Discovery Request (PADR) packet to the AC to which it wants to connect.

4. The selected AC sends a PPPoE Active Discovery Session (PADS) packet to confirm the session.

Session

When Discovery is successfully completed, both the host and the selected remote AC have the information they need to build their point-to-point connection over Ethernet.

The only parameter that you can configure is the number of PPPoE sessions.

NOTE: The router supports dynamic PPPoE interfaces. Also, profiles support PPPoE interfaces. See *Chapter 15, Configuring Dynamic Interfaces* and *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*, for more information.

PPPoE Service Name Tables

PPPoE clients use service name tags, as defined in RFC 2516, to request that an AC support certain services. The client includes a specific service name tag in the PADI packet that it broadcasts to remote ACs, or it can include an empty service name tag of zero length to indicate that any service is acceptable.

On receipt of a PADI packet that it can serve, the AC responds with a PADO packet. The PADO packet contains a service name tag that is identical to the one in the PADI, as well as one or more additional service name tags indicating other services that the AC offers.

Features

PPPoE service name tables enable an AC, such as an E-series router, to support multiple service name tags in addition to the empty service name tag. You can configure up to 16 different PPPoE service name tables per E-series router to:

- Define the set of specific service name tags that the router advertises in the PADO packets sent to PPPoE clients.

- Control whether the router responds to (terminate) or ignores (drop) PADI requests containing an empty service name tag.

Table Structure

Each entry, or row, in a PPPoE service name table consists of the following components:

- Service name tag—Service name tags specify the client services that an AC supports. Every PPPoE service name table includes one empty service name tag, which is a tag of zero length used to represent any service. In addition, you can configure up to 16 specific service name tags per table to specify custom values such as an ISP name or class of service.

- Action—Each service name tag has an associated action: terminate (the default action) or drop. For the empty service name tag, you can specify that the router ignore (drop), rather than respond to (terminate), all PADI requests containing the empty service name tag. By contrast, when you configure a specific (custom) service name tag, you cannot specify the action; the default action, terminate, is always used.

For example, Table 15 shows a PPPoE service name table containing four entries: an empty service name tag (“ ”) associated with the drop action, and three specific service name tags. Note that the only action currently supported for a specific service name tag is terminate.

Table 15: Sample PPPoE Service Name Table

Service-Name	Action
“myISPService”	Terminate
“myQOSClass1”	Terminate
“myQOSClass2”	Terminate
“ ”	Drop

Enabling the Table for Use

After you create a PPPoE service name table and populate it with entries, you must enable it for use with a static or dynamic PPPoE interface. To enable a PPPoE service name table for use with a static interface, you assign the table to the PPPoE major interface. To enable a PPPoE service name table for use with a dynamic interface, you add the table to a profile that is dynamically assigned to a PPPoE interface column. For details about configuring and using PPPoE service name tables, see *Configuring PPPoE Service Name Tables* on page 334.

Using the PPPoE Remote Circuit ID to Identify Subscribers

You can enable the router to capture and format a vendor-specific tag containing a PPPoE remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM) device. The router can then send this value to a Remote Authentication Dial-In User Service (RADIUS) server or to a Layer 2 Tunneling Protocol (L2TP) network server (LNS) to uniquely identify subscriber locations.

This feature is supported on all modules on which you can configure PPPoE interfaces. The feature is particularly useful in Ethernet-based Broadband Remote Access Server (B-RAS) configurations as a means of uniquely identifying subscribers connected to the router on a single Ethernet link.

For detailed configuration instructions, see *Configuring PPPoE Remote Circuit ID Capture* on page 342.

Application

When a connection between an E-series router and a DSLAM is on an ATM interface, subscribers are typically assigned an ATM PVC to communicate with the router. Each ATM PVC is created on a different ATM 1483 subinterface. When a RADIUS server in this configuration sends messages to the router containing the NAS-Port-Id [87] RADIUS attribute, each ATM 1483 subinterface produces a unique NAS-Port-Id that can differentiate subscribers on the ATM link.

By contrast, when the connection between the router and the DSLAM is on an Ethernet interface that does not use either virtual LANs (VLANs) or stacked VLANs (S-VLANs), the NAS-Port-Id value is the same for all subscribers on the Ethernet link. Enabling the router to capture the remote circuit ID sent from the DSLAM and use it as a RADIUS or L2TP attribute facilitates the process of identifying individual subscribers on an Ethernet link.

PPPoE Remote Circuit ID Capture

When you enable capture of the PPPoE remote circuit ID by issuing the **pppoe remote-circuit-id** command, the E-series router captures the remote circuit ID value if it is sent from the DSLAM. The PPPoE intermediate agent on the DSLAM appends a vendor-specific tag containing the remote circuit ID to the existing PPPoE PADI or PADR packet and sends this packet to the E-series router. The PPPoE remote circuit ID value can be a maximum of 64 characters. The router stores this value on the line module on which the PPPoE interface is configured.

PPPoE Remote Circuit ID Format

By default, the router formats the captured PPPoE remote circuit ID to include only the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM. To configure a nondefault format for the captured PPPoE remote circuit ID, you can use one of the **radius remote-circuit-id-format** commands listed in Table 16.

Table 16: Configuring Nondefault Formats for the PPPoE Remote Circuit ID

To Configure This Nondefault Format	Use This Command
Include only the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format agent-remote-id**
Include both the agent-circuit-id suboption (suboption 1) and the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format agent-circuit-id agent-remote-id**
Include the NAS-Identifier [32] RADIUS attribute with either or both of the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent	host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id** or host1(config)#**radius remote-circuit-id-format nas-identifier agent-remote-id** or host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id agent-remote-id**
Append the agent-circuit-id suboption to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006). For details about how the router implements this format, see *Format for dsl-forum-1 Keyword* on page 316.	host1(config)#**radius remote-circuit-id-format dsl-forum-1**

For more information about configuring the format of the PPPoE remote circuit ID value, see **radius remote-circuit-id-format** on page 347.

Remote Circuit ID Delimiter

If the format of the PPPoE remote circuit ID consists of two or more components, the router uses a # character by default to delimit the components. Optionally, you can use the **radius remote-circuit-id-delimiter** command to configure a nondefault delimiter character (for example, ! or %) to separate multiple components in the PPPoE remote circuit ID value. For information about how to use this command, see **radius remote-circuit-id-delimiter** on page 346.

Format for dsl-forum-1 Keyword

When you specify the **radius remote-circuit-id-format** command with the **dsl-forum-1** keyword, the router appends the agent-circuit-id suboption value to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006).

The format of the PPPoE remote circuit ID when you use the **dsl-forum-1** keyword is as follows:

dslForum1InterfaceSpecifier#agent-circuit-id

where:

- *dslForum1InterfaceSpecifier* is the interface specifier in **dsl-forum-1** format
- # is the default delimiter character
- *agent-circuit-id* is the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM

If the DSLAM transmits empty data for *agent-circuit-id*, the router appends the value 0/0/0/0/0/0 to *dslForum1InterfaceSpecifier*.

To obtain the value for *dslForum1InterfaceSpecifier*, the router translates an internally generated interface specifier into the format for the **dsl-forum-1** keyword, using the following conventions:

- The **dsl-forum-1** format for ATM interfaces is atm *slot*/*adapter*/*port*:*vpi*.*vci*
- The **dsl-forum-1** format for Ethernet interfaces is eth *slot*/*adapter*/*port*:*svlanId*.*vlanId*
- For the E120 router or the E320 router, the router uses the actual *adapter* value (0 or 1) in the **dsl-forum-1** format. For ERX-14xx models, ERX-7xx models, and the ERX-310 router, which do not support an *adapter* value, the router sets the *adapter* value to 0 (zero).
- For Ethernet interfaces that use VLANs but do not use S-VLANs, the router sets the *svlanId* value to 4096 and uses the actual *vlanId* value in the **dsl-forum-1** format.
- For Ethernet interfaces that use neither S-VLANs nor VLANs, the router sets both the *svlanId* value and the *vlanId* value to 4096 in the **dsl-forum-1** format.
- The router ignores subinterface values for ATM and Ethernet interfaces in the translated **dsl-forum-1** format.

NOTE: The format of the interface specifier that the router generates internally is different from the interface specifier format that you use to configure interfaces on the router. For information about the interface types and specifiers to use when configuring interfaces on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Format Examples for dsl-forum-1 Keyword

Table 17 provides several examples of how the router uses the conventions described in *Format for dsl-forum-1 Keyword* on page 316 to translate internally generated interface specifiers into the format of the *dslForum1InterfaceSpecifier* value. The examples in the table use adapter 1 for interfaces on an E120 router or E320 router, and adapter 0 (no adapter value) for interfaces on ERX-14xx models, ERX-7xx models, and the ERX-310 router.

Table 17: Interface Specifier Format Examples for dsl-forum-1 Keyword

Interface Example	Internal Router Format	How Router Translates	Format of *dslForum1InterfaceSpecifier*
ATM 1483 subinterface on slot 2, port 0, subinterface 1 with VPI 100 and VCI 101	atm 2/0.1:100.101	■ Sets *adapter* to 0 ■ Ignores subinterface 1 ■ Uses other values as supplied	atm 2/0/0:100.101
ATM 1483 subinterface on slot 3, adapter 1, port 7, subinterface 6 with VPI 200 and VCI 201	atm 3/1/7.6:200.201	■ Ignores subinterface 6 ■ Uses other values as supplied	atm 3/1/7:200.201
Gigabit Ethernet interface on slot 2, port 0 with no VLAN or S-VLAN subinterfaces	gigabitEthernet 2/0	■ Sets *adapter* to 0 ■ Sets both *svlanId* and *vlanId* to 4096 ■ Uses other values as supplied	eth 2/0/0:4096.4096
Gigabit Ethernet interface on slot 4, adapter 1, port 1 with no VLAN or S-VLAN subinterfaces	gigabitEthernet 4/1/1	■ Sets both *svlanId* and *vlanId* to 4096 ■ Uses other values as supplied	eth 4/1/1:4096.4096
Gigabit Ethernet interface on slot 2, port 0, subinterface 1 with VLAN ID 5	gigabitEthernet 2/0.1:5	■ Sets *adapter* to 0 ■ Ignores subinterface 1 ■ Sets *svlanId* to 4096 ■ Uses other values as supplied	eth 2/0/0:4096.5
Gigabit Ethernet interface on slot 4, adapter 1, port 1, subinterface 3 with VLAN ID 10	gigabitEthernet 4/1/1.3:10	■ Ignores subinterface 3 ■ Sets *svlanId* to 4096 ■ Uses other values as supplied	eth 4/1/1:4096.10
Gigabit Ethernet interface on slot 2, port 0, subinterface 1 with S-VLAN ID 5 and VLAN ID 6	gigabitEthernet 2/0.1:5-6	■ Sets *adapter* to 0 ■ Ignores subinterface 1 ■ Replaces - (hyphen) between *svlanId* and *vlanID* with . (period) ■ Uses other values as supplied	eth 2/0/0:5.6
Gigabit Ethernet interface on slot 4, adapter 1, port 1, subinterface 3 with S-VLAN ID 10 and VLAN ID 20	gigabitEthernet 4/1/1.3:10-20	■ Ignores subinterface 3 ■ Replaces - (hyphen) between *svlanId* and *vlanID* with . (period) ■ Uses other values as supplied	eth 4/1/1:10.20

Use by RADIUS or L2TP

Enabling the router to capture and format the PPPoE remote circuit ID sent from the DSLAM has no effect by itself. To use the PPPoE remote circuit ID value, you must send it to a RADIUS server, to an L2TP network server (LNS), or to both by doing one or more of the following:

- Issue the **radius override calling-station-id remote-circuit-id** command to substitute the remote circuit ID value for the standard Calling-Station-Id [31] RADIUS attribute.
- Issue the **radius override nas-port-id remote-circuit-id** command to substitute the remote circuit ID value for the standard NAS-Port-Id [87] RADIUS attribute.
- Issue the **aaa tunnel calling-number-format** command to generate L2TP Calling Number attribute value pair (AVP) 22 in a descriptive format that includes either or both of the agent-circuit-id (suboption 1) and agent-remote-id (suboption 2) suboptions of the PPPoE intermediate agent tags.

For more information about configuring RADIUS and L2TP on E-series routers, see the *JUNOSe Broadband Access Configuration Guide*.

System Event Log

You can use the radiusSendAttributes system event log category to troubleshoot applications that use PPPoE remote circuit ID capture. The radiusSendAttributes event category logs RADIUS attributes added to outbound RADIUS requests.

You can also use the **log severity debug pppoeControlPacket** command to configure a packet trace log for a PPPoE interface that includes the PPPoE remote circuit ID value captured on that interface. For information about how to use the **log severity debug pppoeControlPacket** command, see *Troubleshooting* on page 361.

For information about how to log system events, see *JUNOSe System Event Logging Reference Guide, Chapter 1, System Logging Overview*.

PPPoE MTU Configuration

To avoid fragmentation and reassembly, Ethernet access networks require larger MTU sizes for PPP traffic. With JUNOSe PPPoE MTU, you can control the deployment of larger packet sizes. You can configure PPPoE MTU directly on the PPPoE interface or use a dynamic configuration profile. When you use the PPPoE MTU tag, each PPPoE subinterface can have a unique MTU value. Operational MTU is the lesser of the PPPoE MTU or the lower layer MTU minus the PPPoE overhead.

You can use the **pppoe mtu** command to set the MTU using a combination of lower layer restrictions and controls:

- Greater MTU than the current maximum permitted by RFC 2516, with the default equal to the current maximum setting (1494 octets)
- Optional setting for absolute maximum PPPoE MTU
- Optional use of a larger lower layer MTU
- Optional use of the PPPoE-Max-Mtu tag transmitted from the client

Platform Considerations

You can configure PPPoE interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support PPPoE interfaces on ERX-14xx models, ERX-7xx 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 PPPoE.

For information about the modules that support PPPoE interfaces 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 PPPoE.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the physical interface on which you want to configure PPPoE. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about PPPoE, consult the following resources:

- DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006)
- Extensions to a Method for Transmitting PPP over Ethernet (PPPoE)—draft-carrel-info-pppoe-ext-00.txt (November 2000 expiration)
- IEEE 802.1q (Virtual LANs)
- RFC 2516—Method for Transmitting PPP over Ethernet (PPPoE) (February 1998)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Before You Configure PPPoE

Before you attempt to configure a PPPoE interface, configure the physical interface over which PPPoE traffic will flow. The procedures described in this chapter assume that a physical interface has been configured.

Configuring PPPoE over ATM

This section provides an example of a common PPPoE over ATM configuration.

See the following resources for additional information:

- *Chapter 1, Configuring ATM*—Provides detailed information about ATM technology and line interface module capabilities.
- *Chapter 12, Configuring Bridged Ethernet*—Provides configuration information about Bridged Ethernet, which allows multiple upper-layer interface types (IP and PPPoE) to be simultaneously multiplexed over the same interface.

- *Chapter 15, Configuring Dynamic Interfaces*—Provides detailed information about configuring ATM to support dynamic interfaces.
- *Chapter 4, Configuring Upper-Layer Protocols over Static Ethernet Interfaces*

To configure PPPoE over ATM:

1. Configure a physical interface.

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

2. Configure the ATM 1483 subinterface.

   host1(config-if)#**interface atm 0/1.20**

3. Configure a PVC by specifying the *vcd* (virtual circuit descriptor), the *vpi* (virtual path identifier), the *vci* (virtual channel identifier), and the encapsulation type.

   host1(config-if)#**atm pvc 10 22 100 aal5snap**

4. Select PPPoE as the encapsulation method.

   host1(config-subif)#**encapsulation pppoe**

5. Configure a maximum number of PPPoE sessions on the interface.

   host1(config-if)#**pppoe sessions 128**

6. Create a PPPoE subinterface.

   host1(config-subif)#**interface atm 0/1.20.1**

7. Select PPP as the encapsulation method.

   host1(config-subif)#**encapsulation ppp**

8. (Optional) Configure maximum transfer unit (MTU) parameters.

   host1(config-if)#**pppoe mtu 1380**

9. (Optional) Configure an access concentrator (AC) name on the PPPoE interface.

   host1(config-subif)#**pppoe acname CYM9876**

10. (Optional) Set up the router to prevent a client from establishing more than one session using the same MAC address.

    host1(config-subif)#**pppoe duplicate-protection**

11. Assign an IP address and subnet mask to the PVC.

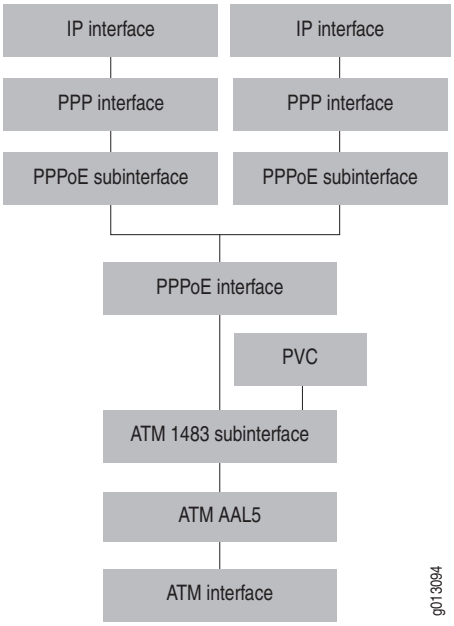
    host1(config-subif)#**ip address 192.32.10.20 255.255.255.0**

12. (Optional) Configure additional PPPoE subinterfaces by completing Steps 6 through 11 using unique numbering.

    host1(config-subif)#**interface atm 0/1.20.2**

Figure 34 illustrates the interface stack for this configuration.

Figure 34: Example of PPPoE over ATM Stacking



atm pvc

- Use to configure a PVC on an ATM interface.
- The following parameters are mandatory:
  - *vcd*—Virtual circuit descriptor, which identifies a virtual circuit in the range 1–2147483647. The *vcd* is a unique number that you assign, which identifies a virtual circuit. The *vcd value* has no relationship to the *vpi* and *vci* values and has meaning only to the E-series router.
  - *vpi*—Virtual path identifier of the PVC. The VPI is an 8-bit field in the ATM cell header. The VPI value is unique on a single link, not throughout the ATM network, because it has meaning only to the E-series router. The VPI value must match the value on the switch. The parameters *vpi* and *vci* cannot be both set to 0; if one is 0, the other cannot be 0.
  - *vci*—Virtual channel identifier. The VCI is a 16-bit field in the ATM cell header. The VCI value is unique on a single link, not throughout the ATM network, because it has meaning only to the E-series router. The parameters *vpi* and *vci* cannot be both set to 0; if one is 0, the other cannot be 0.
  - encapsulation type:
    - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit. An LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
    - **aal5mux ip**—Specifies a multiplexed circuit used for IP only.
    - **aal5autoconfig**—Enables the autodetection of a 1483 encapsulation (LLC/SNAP or VC multiplexed).
- Example

  host1(config-if)#**atm pvc 10 100 22 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

encapsulation ppp

- Use to specify PPP as the encapsulation method for the interface.
- Example

  host1(config-subif)#**encapsulation ppp**

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

encapsulation pppoe

- Use to specify PPPoE as the encapsulation method for the interface.
- Example

  host1(config-subif)#**encapsulation pppoe**

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

interface atm

- Use to configure an ATM interface.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config)#**interface atm 0/1.19**
  host1(config)#**interface atm 0/0/1.19**

- Use the **no** version to remove the interface or subinterface.

ip address

- Use to assign an IP address and subnet mask to a subinterface.
- Example

  host1(config-if)#**ip address 192.1.1.1 255.255.255.0**

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

pppoe acName

- Use to configure an access concentrator (AC) name on the PPPoE interface. When the AC (the server) receives a PPPoE Active Discovery Initiation (PADI) packet that it can serve, it replies by sending a PPPoE Active Discovery Offer (PADO) packet. The PADO packet contains the AC name configured using this command.
- If the AC name is not configured, the router name is used.
- The AC name can be a maximum of 64 characters.

- Example

  host1(config-subif)#**pppoe acName CYM9876**

- Use the **no** version to remove the AC name.

pppoe duplicate-protection

- Use to prevent a client from establishing more than one session using the same MAC address.
- This feature is disabled by default.
- Example

  host1(config-subif)#**pppoe duplicate-protection**

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

pppoe mtu

- Use to set the MTU using a combination of lower layer restrictions and controls.
- You can specify an MTU greater than the current maximum permitted by RFC 2516, in the range 66–65535.
- You can use the **use-lower-layer** keyword to use the lower layer interface value minus any PPPoE overhead. You can use the **use-mtu-tag** keyword to use the provided PPPoE mtu tag value.
- Example

  host1(config-profile)#**pppoe mtu 1380**

- Use the **no** version to restore the default value, 1494.

pppoe sessions

- Use to specify the maximum number of PPPoE subinterfaces permitted on an interface, in the range 1–8000 (ERX routers) or 1–16,000 (E120 and E320 routers). The default value is 8000 (ERX routers) or 16,000 (E120 and E320 routers).
- The **pppoe sessions** command affects only those subinterfaces that you create after issuing this command. Previously created interfaces remain, even if their number exceeds the new value for **pppoe sessions**.
- Example

  host1(config-if)#**pppoe sessions 128**

- Use the **no** version to restore the default value, 8000 (ERX routers) or 16,000 (E120 and E320 routers).

Configuring PPPoE for Ethernet Modules

You can configure PPPoE on Fast Ethernet (FE), Gigabit Ethernet (GE), and 10-Gigabit Ethernet (10GE) modules. You can configure Ethernet interfaces with IP only, with PPPoE only, with both IP and PPPoE, and with or without VLANs.

This section provides information about configuring PPPoE without VLANs. If you want to configure PPPoE with VLANs, see *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces*, which shows common VLAN configurations such as:

- PPPoE over VLAN
- IP over VLAN and PPPoE over VLAN

NOTE: *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces* provides other non-VLAN configuration examples, such as configurations using MPLS.

For more information about specific Ethernet modules and the protocols and applications they support, see:

- *ERX Module Guide*, *Appendix A*, *Module Protocol Support* (for ERX-7xx models, ERX-14xx models, and ERX-310 routers)
- *E120 and E320 Module Guide*, *Appendix A*, *IOA Protocol Support* (for E120 routers and E320 routers)

PPPoE Interface and Subinterface Limits

PPPoE subinterfaces can be distributed in any way across I/O module ports. For example, you can configure the maximum supported number of PPPoE subinterfaces on one port of an FE-2 I/O module and no PPPoE subinterfaces on the other port.

For information about current system maximums supported for PPPoE interfaces and subinterfaces, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

Configuring PPPoE Without VLANs

To configure PPPoE over an Ethernet interface without VLANs:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify PPPoE as the encapsulation method on the interface.

   host1(config-if)#**pppoe**

3. Create a PPPoE subinterface.

   host1(config-if)#**pppoe subinterface fastEthernet 4/1.1**

4. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**

5. (Optional) Configure an access concentrator (AC) name on the PPPoE interface.

   host1(config-subif)#**pppoe acname CYM9876**

6. (Optional) Set up the router to prevent a client from establishing more than one session using the same MAC address.

   host1(config-subif)#**pppoe duplicate-protection**

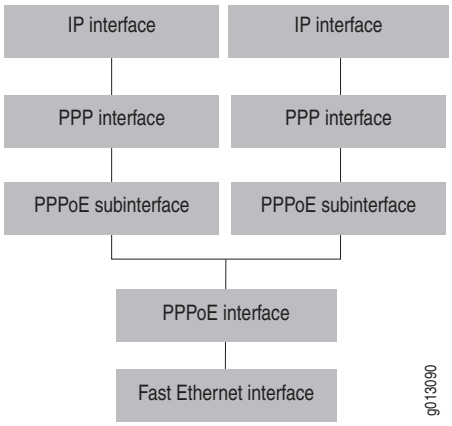
7. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

8. (Optional) Configure additional PPPoE subinterfaces by completing Steps 3 through 7 using unique numbering.

Figure 35 illustrates the interface stack for this configuration.

Figure 35: Example of PPPoE Stacking



encapsulation ppp

- Use to specify PPP as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation ppp**

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

interface fastEthernet

- Use to select a Fast Ethernet interface.
- For more information, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.
- 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 subinterface if the one above it still exists.

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- To specify a Gigabit Ethernet interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format.
- To specify a Gigabit Ethernet interface or 10-Gigabit Ethernet interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
- For more information, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

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

ip address

- Use to assign an IP address and subnet mask to an interface or subinterface.
- Example

  host1(config-if)#**ip address 192.1.1.1 255.255.255.0**

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

pppoe

- Use to specify PPPoE as the encapsulation method for the interface.
- This command creates a PPPoE major interface.
- Example

  host1(config-if)#**pppoe**

- Use the **no** version to remove the PPPoE major interface.

pppoe acName

- Use to configure an access concentrator (AC) name on the PPPoE interface. When the AC (the server) receives a PPPoE Active Discovery Initiation (PADI) packet that it can serve, it replies by sending a PPPoE Active Discovery Offer (PADO) packet. The PADO packet contains the AC name configured using this command.
- If the AC name is not configured, the router name is used.
- The AC name can be a maximum of 64 characters.
- Example

  host1(config-subif)#**pppoe acName CYM9876**

- Use the **no** version to remove the AC name.

pppoe duplicate-protection

- Use to prevent a client from establishing more than one session using the same MAC address.
- This feature is disabled by default.
- Example

  host1(config-subif)#**pppoe duplicate-protection**

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

pppoe subinterface fastEthernet

- Use to create a PPPoE subinterface on a Fast Ethernet module.
- On ERX-7xx models, ERX-14xx models, and the ERX-310 router, use the *slot*/*port*/*pppoeSubinterface* format.
- Example

  host1(config)#**pppoe subinterface fastEthernet 4/1.1**

- Use the **no** version to remove the PPPoE subinterface.

Configuring PADM Messages

You can configure PPPoE to issue and display a PPPoE Active Discovery Message (PADM). The PADM message is a control message that servers send to clients. The clients may act on the control message, but are not required to do so. There are two types of PADM messages:

- Message of the minute (MOTM)—Informs clients of interesting system information
- URL—Typically spawns an Internet browser with the specified URL as the initial page

You can configure the router to send PADM messages as follows:

- Send MOTM messages to all clients connected to the router.
- Send MOTM and URL messages to all clients connected to a subinterface.
- Configure profiles to send MOTM and URL messages to new clients created when the profile is dynamically attached to an IP interface.

NOTE: You can use the **pppoe motm** command at three different points in the configuration process: Privileged Exec, Interface Configuration, and Profile Configuration modes. You can use the **pppoe url** command at two different points in the configuration process: Interface Configuration and Profile Configuration modes. Note the differences described in guidelines below.

pppoe motm

- Use to cause the PPPoE application to send a PADM message of the minute (MOTM) message to all PPPoE clients connected to the router. The MOTM string is passed with no changes.
- The message string is not saved in nonvolatile storage (NVS).
- Use in Privileged Exec mode.
- Example

  host1#**pppoe motm Router going down at 10:00 p.m.**

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

pppoe motm

- Use in the context of a PPPoE subinterface to cause the PPPoE application to send the specified PADM message to the client as it is configured (if connected).
- The message is also sent whenever the subinterface transitions from down to up.
- The message string is saved in nonvolatile storage (NVS).
- Use in Interface Configuration mode.
- Example

  host1(config-if)#**interface fastEthernet 1/0.1.1**
  host1(config-if)#**pppoe motm Router going down at 10:00 p.m.**

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

pppoe motm

- Use in a profile to cause the PPPoE application to send the string to the new client that is created when the profile is dynamically attached to an IP interface.
- The message string is saved in nonvolatile storage (NVS).
- Use in Profile Configuration mode.

- Example

  host1(config-profile)#**pppoe motm Router going down now**

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

pppoe url

- Use in the context of a PPPoE subinterface to cause the PPPoE application to send the specified PADM message to the client as it is configured (if connected).
- The message is also sent whenever the subinterface transitions from down to up.
- The message string is saved in nonvolatile storage (NVS).
- Use in Interface Configuration mode.
- Example

  host1(config-if)#**interface fastEthernet 1/0.1.1**
  host1(config-if)#**pppoe url http://www.relevanturl.com**

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

pppoe url

- Use in a profile to cause the PPPoE application to send the string to the new client that is created when the profile is dynamically attached to an IP interface.
- The message string is saved in nonvolatile storage (NVS).
- PPPoE substitutes the following characters for information in the specified URL string before transmitting:
  - %U user and domain name
  - %u user name
  - %d domain name
  - %D profile name
  - %% % character
- Use in Profile Configuration mode.
- Example

  host1(config-profile)#**pppoe url http://www.relevanturl.com**

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

Configuring PADN Messages

You can configure PPPoE to receive PPPoE Active Discovery Network (PADN) messages. When a client connects to a PPPoE server, such as an E-series router, the client receives configuration information from the server via the PADN message. This PADN information associates the PPPoE sessions with a set of routes. The client can use this set of routes to determine which session to use based on the destination IP address.

The PADN packet data is relevant only when the PPP network layer is "up." To reach an up state, PPP alerts PPPoE after the Network Control Protocol (NCP) completes negotiation.

The routes of interest can be maintained on the router in domain maps.

NOTE: For information about domain mapping, see *JUNOSe Broadband Access Configuration Guide, Chapter 1, Configuring Remote Access*.

aaa domain-map

- Use to map a domain name between a PPP client's domain name and a virtual router.
- Example

  host1(config)#**aaa domain-map xyz.com**
  host1(config-domain-map)#**padn 10.2.25.6 255.255.255.0 10**
  host1(config-domain-map)#**padn 20.2.0.0 255.255.0.0 11**

- Use the **no** version to delete the map entry.

padn

- Use to configure PADN parameters for a domain name.
- You may send up to a maximum of 16 PADNs per domain name.
- Example

  host1(config-domain-map)#**padn 10.2.25.6 255.255.255.255 13**

- Use the **no** version to delete PADN parameters for the domain name.

Configuring PPPoE Service Name Tables

To configure PPPoE service name tables on the router:

1. Create the PPPoE service name table.
2. (Optional) Add entries to populate the PPPoE service name table. You can:
   - Configure specific service names to represent custom values.
   - Specify a nondefault action for the empty service name entry.
3. Enable the PPPoE service name table for use with a static or dynamic interface.

The following sections describe how to perform these tasks.

Creating and Populating PPPoE Service Name Tables

To create and populate a PPPoE service name table on the router:

1. From Global Configuration mode, create a PPPoE service name table by assigning it a name.

   host1(config)#**pppoe-service-name-table myServiceTable1**

   This command accesses PPPoE Service Name Table Configuration mode and builds a default PPPoE service name table named myServiceTable1. The table contains a single empty service name entry associated with the default action, terminate, as shown in Table 18. With no further service name entries, this table directs the router to respond to all PADI requests containing an empty service name tag.

   Table 18: Default PPPoE Service Name Table

Service-Name	Action
" "	Terminate

2. (Optional) From PPPoE Service Name Table Configuration mode, create entries to populate the PPPoE service name table. You can configure up to 16 specific service name entries per table, or modify the action for the empty service name tag.

   host1(config-pppoe-service-name-table)#**service myISPService**
   host1(config-pppoe-service-name-table)#**service myQOSClass1**
   host1(config-pppoe-service-name-table)#**service myQOSClass2**
   host1(config-pppoe-service-name-table)#**service empty-service-name drop**

These commands build the PPPoE service name table shown in Table 19. This table directs the router to send a PADO packet in response to all PADI requests containing the myISPService, myQOSClass1, or myQOSClass2 service name tag, and to ignore (drop) all PADI requests containing empty service name tags.

Table 19: PPPoE Service Name Table with Entries

Service-Name	Action
"myISPService"	Terminate
"myQOSClass1"	Terminate
"myQOSClass2"	Terminate
" "	Drop

3. Exit PPPoE Service Name Table Configuration mode.

   host1(config-pppoe-service-name-table)#**exit**

4. (Optional) Use the appropriate **show** command to verify the creation of the PPPoE service name table and entries.

   host1(config)#**show pppoe-service-name-table name myServiceTable1**

5. (Optional) Repeat Steps 1 through 4 to configure additional PPPoE service name tables on the router.

pppoe-service-name-table

- Use from Global Configuration mode to create a PPPoE service name table.
- You can create a maximum of 16 PPPoE service name tables per E-series router.
- Specify a table name of up to 31 alphanumeric characters.
- This command accesses PPPoE Service Name Table Configuration mode, which enables you to configure entries for the PPPoE service name table.
- Example

  host1(config)#**pppoe-service-name-table myServiceTable1**

- Use the **no** version to remove the specified PPPoE service name table from the router.

service

- Use to add a specific service name tag to a PPPoE service name table, or to modify the action for the empty service name tag in a PPPoE service name table.
- Each PPPoE service name table includes one empty service name tag, and can optionally include up to 16 additional specific service name entries.
- For each specific service name tag that you configure, assign a name of up to 31 alphanumeric characters.
- You cannot configure the action for a specific service name tag; the default action, terminate, is always used.

- For an empty service name tag, you can specify that the AC, such as an E-series router, ignore (drop), rather than respond to (terminate), all PADI requests from the client that contain an empty service name tag.
- Examples

  host1(config-pppoe-service-name-table)#**service myISPService**
  host1(config-pppoe-service-name-table)#**service empty-service-name drop**

- Use the **no** version to restore the default action, terminate, for an empty service name tag, or to remove the specified non-empty service name tag from the PPPoE service name table.

Enabling PPPoE Service Name Tables for Use with Static Interfaces

To enable a PPPoE service name table for use with a static interface, assign the service name table to the PPPoE major interface.

PPPoE over ATM Configurations

To enable a PPPoE service name table for use with a static interface in PPPoE over ATM configurations:

1. Configure an ATM physical interface.

   host1(config)#**interface atm 3/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 3/0.1**

3. Configure an ATM PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**

4. Select PPPoE as the encapsulation method on the interface. This command creates the PPPoE major interface.

   host1(config-subif)#**encapsulation pppoe**

5. Assign the PPPoE service name table to the PPPoE major interface.

   host1(config-subif)#**pppoe service-name-table myServiceTable1**

atm pvc

- Use to configure a PVC on an ATM interface.
- For details about specifying the mandatory VCD, VPI, VCI, and encapsulation type parameters, see **atm pvc** on page 324.
- Example

  host1(config-if)#**atm pvc 10 100 22 aal5snap**

- Use the **no** version to remove the specified PVC.

encapsulation pppoe

- Use to specify PPPoE as the encapsulation method for the interface.
- This command creates a PPPoE major interface.
- Example

  host1(config-subif)#**encapsulation pppoe**

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

interface atm

- Use to configure an ATM interface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 325.
- Examples

  host1(config)#**interface atm 3/1.19**
  host1(config)#**interface atm 3/0/1.19**

- Use the **no** version to remove the interface or subinterface.

pppoe service-name-table

- Use from Subinterface Configuration mode to assign a PPPoE service name table to a PPPoE major interface for use by a static ATM 1483 subinterface.
- Specify the name of the PPPoE service name table configured with the **pppoe-service-name-table** command from Global Configuration mode.
- Example

  host1(config-subif)#**pppoe service-name-table myServiceTable1**

- Use the **no** version to remove the PPPoE service name table assignment.

PPPoE over Ethernet Configurations

To enable a PPPoE service name table for use with a static interface in PPPoE over Ethernet configurations:

1. Configure a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet physical interface.

   host1(config)#**interface fastEthernet 4/1**

2. Select PPPoE as the encapsulation method on the interface. This command creates the PPPoE major interface.

   host1(config-if)#**pppoe**

3. Assign the PPPoE service name table to the PPPoE major interface.

   host1(config-if)#**pppoe service-name-table myServiceTable1**

interface fastEthernet

- Use to select a Fast Ethernet interface.
- Example

  host1(config)#**interface fastEthernet 4/1**

- 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 subinterface if the one above it still exists.

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- For information about specifying the Gigabit Ethernet or 10-Gigabit Ethernet interface or subinterface, see **interface gigabitEthernet** and **interface tenGigabitEthernet** on page 329.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

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

pppoe

- Use to specify PPPoE as the encapsulation method for the interface.
- This command creates a PPPoE major interface.
- Example

  host1(config-if)#**pppoe**

- Use the **no** version to remove the PPPoE major interface.

pppoe service-name-table

- Use from Interface Configuration mode to assign a PPPoE service name table to a PPPoE major interface for use by a static Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.
- Specify the name of the PPPoE service name table configured with the **pppoe-service-name-table** command from Global Configuration mode.
- Example

  host1(config-if)#**pppoe service-name-table myServiceTable1**

- Use the **no** version to remove the PPPoE service name table assignment.

Enabling PPPoE Service Name Tables for Use with Dynamic Interfaces

To enable a PPPoE service name table for use with a dynamic interface, add the service name table to a profile that is dynamically assigned to the interface.

For complete details, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

To enable a PPPoE service name table for use with a dynamic interface:

1. Create a profile by assigning it a name.

   host1(config)#**profile baseProfile**

2. Assign the PPPoE service name table to the profile as a PPPoE characteristic.

   host1(config-profile)#**pppoe service-name-table myServiceTable1**

3. Exit Profile Configuration mode.

   host1(config-profile)#**exit**

4. Configure a physical interface.

   On ERX-7xx models, ERX-14xx models, and the ERX-310 router:

   host1(config-if)#**interface atm 3/0.1**

5. Configure an ATM PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**

6. Apply the profile to the interface.

   host1(config-subif)#**profile pppoe baseProfile**

7. Enable the PPPoE dynamic encapsulation type.

   host1(config-subif)#**auto-configure pppoe**

atm pvc

- Use to configure a PVC on an ATM interface.
- For details about specifying the mandatory VCD, VPI, VCI, and encapsulation type parameters, see **atm pvc** on page 324.
- Example

  host1(config-if)#**atm pvc 10 100 22 aal5snap**

- Use the **no** version to remove the specified PVC.

auto-configure

- Use to configure an ATM 1483 subinterface to support a dynamic interface. Specifies the type(s) of dynamic encapsulation that will be accepted/detected by the ATM 1483 subinterface.
- This command causes the layers above ATM 1483 to become dynamic.
- Select **pppoe** as the dynamic next upper interface type.
- Example

  host1(config-subif)#**auto-configure pppoe**

- Use the **no** version to disable detection of the specified encapsulation.

interface atm

- Use to configure an ATM interface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 325.
- Examples

  host1(config)#**interface atm 3/0.1**
  host1(config)#**interface atm 3/0/0.1**

- Use the **no** version to remove the interface or subinterface.

pppoe service-name-table

- Use from Profile Configuration mode to assign a PPPoE service name table to a profile for use by the dynamic PPPoE interface column associated with the profile.
- Specify the name of the PPPoE service name table configured with the **pppoe-service-name-table** command from Global Configuration mode.
- Example

  host1(config-profile)#**pppoe service-name-table myServiceTable1**

- Use the **no** version to remove the PPPoE service name table assignment.

profile

- Use from Global Configuration mode to create a profile name of up to 80 characters.
- Use from Subinterface Configuration mode to assign a profile to an interface. Specify **pppoe** as the encapsulation type to which the profile applies.
- Examples

  host1(config)#**profile myProfile**
  host1(config-subif)#**profile pppoe myProfile**

- Use the **no** version to remove a profile (from Global Configuration mode) or to remove the profile assignment (from Subinterface Configuration mode).

Configuring PADS Packet Content

By default, an E-series router acting as an AC sends both the AC-Name and AC-Cookie tags as part of the PADS packet when it confirms a session with a PPPoE client. These tags are defined in RFC 2516 as follows:

- AC-Name—String that uniquely identifies the particular AC
- AC-Cookie—Tag used by the AC to help protect against denial of service (DoS) attacks

If necessary for compatibility with your network equipment, you can issue the **pppoe pads disable-ac-info** command to prevent the router from sending the AC-Name and AC-Cookie tags in the PADS packet.

pppoe pads disable-ac-info

- Use to prevent the router from sending the AC-Name and AC-Cookie tags in the PADS packet.
- The **pppoe pads disable-ac-info** command affects PADS packets sent only on PPPoE interfaces configured on the router after the command is issued. It has no effect on PADS packets sent on previously created PPPoE interfaces.
- Example

  host1(config)#**pppoe pads disable-ac-info**

- Use the **no** version to restore the default behavior, which is to send the AC-Name and AC-Cookie tags in the PADS packet.

Configuring PPPoE Remote Circuit ID Capture

To capture and use the PPPoE remote circuit ID:

1. Configure a static or dynamic PPPoE interface.

   For instructions on configuring a static PPPoE interface, see *Configuring PPPoE over ATM* on page 321 or *Configuring PPPoE for Ethernet Modules* on page 327.

   For instructions on configuring a dynamic PPPoE interface, see *Chapter 15, Configuring Dynamic Interfaces*.

2. Configure capture of the PPPoE remote circuit ID on this interface.

   a. Enable the router to capture the PPPoE remote circuit ID transmitted from the DSLAM by using one of the following methods:

      - For a static PPPoE interface, issue the **pppoe remote-circuit-id** command from Interface Configuration mode or Subinterface Configuration mode.

        host1(config-if)#**pppoe remote-circuit-id**

      - For a dynamic PPPoE interface, issue the **pppoe remote-circuit-id** command from Profile Configuration mode as a characteristic of the profile assigned to the dynamic PPPoE interface column.

        host1(config)#**profile pppoeTest**
        host1(config-profile)#**pppoe remote-circuit-id**

      By default, the router formats the captured PPPoE remote circuit ID to include only the agent-circuit-id suboption (suboption 1) of the PPPoE intermediate agent tags sent from the DSLAM.

   b. (Optional) Use the **show pppoe interface** command (for static PPPoE interfaces) or the **show profile** command (for dynamic PPPoE interfaces) to verify that PPPoE remote circuit capture is enabled.

      host1#**show pppoe interface fastEthernet 4/1.1**
      host1#**show profile name pppoeTest**

      For information about how to use these commands, see **show pppoe interface** on page 349 and **show profile** on page 357.

3. (Optional) Configure the format of the captured PPPoE remote circuit ID value.

   a. Configure RADIUS to specify a nondefault format for the PPPoE remote circuit ID value.

      - For example, the following command formats the PPPoE remote circuit ID to include only the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent.

        host1(config)#**radius remote-circuit-id-format agent-remote-id**

      - The following command formats the PPPoE remote circuit ID to include the NAS-Identifier [32] RADIUS attribute with both the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent.

        host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id agent-remote-id**

      - The following command formats the PPPoE remote circuit ID to append the agent-circuit-ID suboption to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006). For details about how the router implements this format, see *Format for dsl-forum-1 Keyword* on page 316.

        host1(config)#**radius remote-circuit-id-format dsl-forum-1**

   b. Configure RADIUS to specify a nondefault delimiter character to separate components in the PPPoE remote circuit ID value. (The default delimiter character is #.)

      host1(config)#**radius remote-circuit-id-delimiter %**

   c. Use the **show radius remote-circuit-id format** command and the **show radius remote-circuit-id-delimiter** command to verify the format and delimiter settings for the PPPoE remote circuit ID value.

      host1#**show radius remote-circuit-id-format**
      host1#**show radius remote-circuit-id-delimiter**

      For information about how to use these commands, see **show radius remote-circuit-id-format** on page 361 and **show radius remote-circuit-id-delimiter** on page 361.

4. Send the PPPoE remote circuit ID value to a RADIUS server, to an LNS, or to both.

   a. Configure RADIUS to use the PPPoE remote circuit ID captured from the DSLAM in place of either (or both) of the Calling-Station-Id [31] and NAS-Port-Id [87] RADIUS attributes.

      host1(config)#**radius override calling-station-id remote-circuit-id**
      host1(config)#**radius override nas-port-id remote-circuit-id**

   b. Configure the E-series L2TP access controller (LAC) to generate L2TP Calling Number AVP 22 in fixed format or one of several formats that includes either or both of the agent-circuit-id (suboption 1) and agent-remote-id (suboption 2) suboptions of the tags supplied by the PPPoE intermediate agent.

      host1(config)#**aaa tunnel calling-number-format fixed**

      or

      host1(config)#**aaa tunnel calling-number-format descriptive include-agent-circuit-id include-agent-remote-id**

      or

      host1(config)#**aaa tunnel calling-number-format include-agent-circuit-id**

   c. (Optional) Configure a fallback format for the L2TP Calling Number AVP 22. The fallback format is used only when you have configured the calling number format as anything other than fixed and the PPPoE agent ID is null or unavailable.

      host1(config)#**aaa tunnel calling-number-format fallback fixed**

      or

      host1(config)#**aaa tunnel calling-number-format fallback descriptive**

   d. (Optional) Use the **show radius override** command to verify the override settings configured for RADIUS, and the **show aaa tunnel-parameters** command to verify the parameters configured for L2TP tunnel definitions.

      host1#**show radius override**
      host1#**show aaa tunnel-parameters**

      For information about how to use these commands, see **show radius override** on page 360 and **show aaa tunnel-parameters** on page 349.

aaa tunnel calling-number-format

- Use to configure the format used by the E-series LAC to generate the L2TP Calling Number AVP 22.
- The fixed format is similar to the fixed format of RADIUS attribute 31 (Calling-Station-Id). The LAC uses this format in ICRQ packets that it sends to the LNS.

- Several different descriptive formats include information about the interface and either or both of the suboptions supplied by the PPPoE intermediate agent, agent-circuit-id and agent-remote-id.
- Several simpler formats include only either or both of the PPPoE suboptions, agent-circuit-id and agent-remote-id.
- Example 1

  host1(config)#**aaa tunnel calling-number-format descriptive include-agent-circuit-id**

- Example 2

  host1(config)#**aaa tunnel calling-number-format descriptive include-agent-circuit-id include-agent-remote-id**

- Example 3

  host1(config)#**aaa tunnel calling-number-format include-agent-circuit-id**

- Use the **no** version to restore the default calling number format, descriptive.

aaa tunnel calling-number-format-fallback

- Use to configure the fallback format that the E-series LAC uses to generate the L2TP Calling Number AVP 22 in the event that the PPPoE agent ID is null or unavailable.
- The fallback format is used only when the configured calling number format includes either or both of the agent-circuit-id and agent-remote-id.
- The calling number format determines what element triggers use of the fallback format:

Calling Number Format	Fallback Trigger
agent-circuit-id	agent-circuit-id is empty
agent-circuit-id include-agent-remote-id	Both agent-circuit-id and agent-remote-id are empty.
agent-remote-id	agent-remote-id is empty
descriptive include-agent-circuit-id	agent-circuit-id is empty
descriptive include-agent-circuit-id include-agent-remote-id	Both agent-circuit-id and agent-remote-id are empty.
descriptive include-agent-remote-id	agent-remote-id is empty

- You can specify either descriptive format or fixed format.
- Example

  host1(config)#**aaa tunnel calling-number-format-fallback fixed**

- Use the **no** version to restore the default format, descriptive.

pppoe remote-circuit-id

- Use to enable a static PPPoE interface (from Interface Configuration mode or Subinterface Configuration mode) or a dynamic PPPoE interface (from Profile Configuration mode) to capture and process a vendor-specific tag containing a remote circuit ID transmitted from a DSLAM.
- The router can then send this value to a RADIUS server or to an L2TP network server (LNS) to uniquely identify subscriber locations.
- Examples

  host1(config-if)#**pppoe remote-circuit-id**
  host1(config-profile)#**pppoe remote-circuit-id**

- Use the **no** version to restore the default behavior, which is not to capture and process the PPPoE remote circuit ID.

radius override calling-station-id remote-circuit-id

- Use to configure RADIUS to override the standard use of the Calling-Station-Id [31] RADIUS attribute and instead use the PPPoE remote circuit ID transmitted from a DSLAM.
- Example

  host1(config)#**radius override calling-station-id remote-circuit-id**

- Use the **no** version to restore the default Calling-Station-Id value, which is the telephone number from which the call originated.

radius override nas-port-id remote-circuit-id

- Use to configure RADIUS to override the standard use of the NAS-Port-Id [87] RADIUS attribute and instead use the PPPoE remote circuit ID transmitted from a DSLAM.
- Example

  host1(config)#**radius override nas-port-id remote-circuit-id**

- Use the **no** version to restore the default NAS-Port-Id value, which is the physical interface of the network access server (NAS) that is authenticating the user.

radius remote-circuit-id-delimiter

- Use to configure the delimiter character that the router uses to set off multiple components in the format of the PPPoE remote circuit ID value captured from a DSLAM.
- Example

  host1(config)#**radius remote-circuit-id-delimiter !**

- Use the **no** version to restore the default delimiter character, #.

radius remote-circuit-id-format

- Use to configure the format of the PPPoE remote circuit ID value captured from a DSLAM.
- By default, the router formats the PPPoE remote circuit ID to include only the agent-circuit-id suboption (suboption 1) of the tags supplied by the PPPoE intermediate agent.
- You can use this command to configure one of the following nondefault formats for the PPPoE remote circuit ID value:
  - To include the agent-circuit-id suboption, use the **agent-circuit-id** keyword.
  - To include the agent-remote-id suboption (suboption 2) of the tags supplied by the PPPoE intermediate agent, use the **agent-remote-id** keyword.
  - To include the NAS-Identifier [32] RADIUS attribute, use the **nas-identifier** keyword. If you include the **nas-identifier** keyword, you must also include either or both of the **agent-circuit-id** and **agent-remote-id** keywords.
  - To append the agent-circuit-ID value to an interface specifier that is consistent with the recommended format in the DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006), use the **dsl-forum-1** keyword. For details about how the router implements this format, see *Format for dsl-forum-1 Keyword* on page 316.
- RADIUS overrides the standard use of the Calling-Station-Id [31] or NAS-Port-Id [87] attribute with the PPPoE remote circuit ID only if the DSLAM transmits non-empty data for at least one of the agent-circuit-id or agent-remote-id values. If the DSLAM transmits empty data, then RADIUS does not override the Calling-Station-Id [31] or NAS-Port-Id [87] RADIUS attribute with the PPPoE remote circuit ID and instead uses the standard value for the RADIUS attribute.
- If a single component in a multi-component PPPoE remote circuit ID format is empty, the router represents the empty component as two consecutive delimiter characters (## by default).
- Example 1—Formats the PPPoE remote circuit ID value to include only the agent-remote-id suboption of the tags supplied by the PPPoE intermediate agent.

  host1(config)#**radius remote-circuit-id-format agent-remote-id**

- Example 2—Formats the PPPoE remote circuit ID value to include both the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent.

  host1(config)#**radius remote-circuit-id-format agent-circuit-id agent-remote-id**

- Example 3—Formats the PPPoE remote circuit ID value to include the NAS-Identifier [32] RADIUS attribute with the agent-circuit-id suboption of the tags supplied by the PPPoE intermediate agent.

  host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id**

- Example 4—Formats the PPPoE remote circuit ID value to include the NAS-Identifier [32] RADIUS attribute with the agent-remote-id suboption of the tags supplied by the PPPoE intermediate agent.

  host1(config)#**radius remote-circuit-id-format nas-identifier agent-remote-id**

- Example 5—Formats the PPPoE remote circuit ID value to include the NAS-Identifier [32] RADIUS attribute with both the agent-circuit-id and agent-remote-id suboptions of the tags supplied by the PPPoE intermediate agent.

  host1(config)#**radius remote-circuit-id-format nas-identifier agent-circuit-id agent-remote-id**

- Example 6—Formats the PPPoE remote circuit ID value to use the format for the **dsl-forum-1** keyword. For details about how the router implements this format, see *Format for dsl-forum-1 Keyword* on page 316.

  host1(config)#**radius remote-circuit-id-format dsl-forum-1**

- Use the **no** version to restore the default format, agent-circuit-id.

Monitoring PPPoE

Use the commands described in this section to display information about PPPoE interfaces and subinterfaces.

You can set a statistics baseline for PPPoE interfaces, subinterfaces, and circuits using the **baseline pppoe interface** command.

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.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

baseline pppoe interface

- Use to set a statistics baseline for PPPoE interfaces, subinterfaces, and circuits.
- 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.
- You cannot set a baseline for groups of interfaces, subinterfaces, or circuits. You must set them one at a time.
- When baselining is requested, the time since the last baseline was set is displayed in *hours:minutes:seconds* or *days/hours* format. If a baseline has not been set, the message “No baseline has been set” is displayed instead.
- Use the optional **delta** keyword with PPPoE **show** commands to specify that baselined statistics will be shown.
- Examples

  host1#**baseline pppoe interface atm 2/0.1.1**
  host1#**baseline pppoe interface atm 2/0/0.1.1**

- There is no **no** version.

show aaa tunnel-parameters

- Use to display tunnel parameters that are configured for L2TP tunnel definitions, including the calling number format.
- Field descriptions
  - Tunnel password—Default tunnel password
  - Tunnel client-name—Hostname that the LAC sends to the LNS when communicating about the tunnel
  - Tunnel nas-port-method—Default NAS port type
  - Tunnel nas-port ignore—Whether the router uses the tunnel peer's NAS-Port [5] attribute; enabled or disabled
  - Tunnel nas-port-type ignore—Whether the router uses the tunnel peer's NAS-Port-Type [61] attribute; enabled or disabled
  - Tunnel assignmentID format—Value of the tunnel assignment ID that is passed to PPP/L2TP
  - Tunnel calling number format—Format configured for L2TP Calling Number AVP 22 generated by the LAC
- Example

```
host1#show aaa tunnel-parameters
Tunnel password is 3&92k5b#q4
Tunnel client-name is host1
Tunnel nas-port-method is none
Tunnel nas-port ignore disabled
Tunnel nas-port-type ignore disabled
Tunnel assignmentId format is assignmentId
Tunnel calling number format is descriptive, includes agent-circuit-id and
agent-remote-id
```

show pppoe interface

- Use to display parameters on a PPPoE interface or a PPPoE subinterface.
- If you do not specify an interface and subinterface, the router displays the PPPoE interface and Status parameters for all configured interfaces.
- If you specify an interface with no subinterface, the router displays the PPPoE interface and Status parameters for that interface.
- If you specify an interface and subinterface, the router displays detailed parameters available for that subinterface.
- Field descriptions
  - PPPoE interface—Interface identifier. For more information about specifying the physical interface, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - Status—Operational status of the interface; possible values are:
    - operStatusUp—Interface or subinterface is operational
    - Down—Interface or subinterface is not operational
    - LowerLayerDown—*S*ubinterface is not operational because an underlying interface is down

- full—Displays configuration, status, and statistics information
- max sessions—Number of maximum allowable PPP sessions configured
- mtu—Maximum transfer unit (MTU) value; when derived from the PPPoE MTU tag, the value can only be determined from an active session.
- acName—Name of PPPoE access concentrator
- will not send ac info in PADS packet—When the **pppoe pads disable-ac-info** command is issued, indicates that the router does not send the AC-Name and AC-Cookie tags in the PADS packet
- duplicate-protection—Whether duplicate protection is enabled or disabled for the interface
- capture remote circuit id—Whether capture of the PPPoE remote circuit ID sent from the DSLAM is enabled or disabled for the interface
- active connections—Number of live PPP connections
- configured subinterfaces—Number of PPPoE subinterfaces you configured on an interface
- Assigned profile—Name of profile assigned to dynamic PPPoE interface
- PPPoE Statistics Counters
  - PADI received/PADI transmitted—Number of initiation control packets received/transmitted
  - PADO received/PADO transmitted—Number of offer control packets received/transmitted
  - PADR received/PADR transmitted—Number of request control packets received/transmitted
  - PADS received/PADS transmitted—Number of session confirmation control packets received/transmitted
  - PADT received/PADT transmitted—Number of termination control packets received/transmitted
  - PADM received/PADM transmitted—Number of message control packets received/transmitted
  - PADN received/PADN transmitted—Number of network control packets received/transmitted
- PAD packets received—Total number of control packets received on the interface
- PAD packets transmitted—Total number of control packets transmitted on the interface
- Invalid PAD Packets
  - Invalid Version—Number of control packets received with an invalid version
  - Invalid PAD Code—Number of control packets received with an invalid code
  - Invalid PAD Tags—Number of control packets received with invalid tags
  - Invalid PAD Tag length—Number of control packets received with an invalid tag length

- Invalid PAD Type—Number of control packets received with an invalid type
- Invalid PADI Session—Number of invalid PPPoE Active Discovery Initiation sessions
- Invalid PADR Session—Number of invalid PPPoE Active Discovery Request sessions
- Invalid PAD packet length—Number of control packets received with an invalid packet length
- Invalid PAD packets—Number of invalid control packets received

- Total Invalid PAD packets—Total number of invalid control packets received on the interface
- Insufficient Resources—Number of requests denied because of an inadequate number of sessions; check the number of active clients
- Lockout Configuration (seconds)—Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column; for more information about these fields, see *Configuring Encapsulation Type Lockout for PPPoE Clients* in *Chapter 15, Configuring Dynamic Interfaces*
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout
  - Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period

- Example 1

```
host1#show pppoe interface atm 1/0.1
PPPoE interface ATM 1/0.1 is operStatusUp (dynamic)
    PPPoE interface ATM 1/0.1 has max sessions = 4000
    PPPoE interface ATM 1/0.1 has acName of 111111111111111
    PPPoE interface ATM 1/0.1 will not send ac info in PADS packet
    PPPoE interface ATM 1/0.1 is in duplicate-protection
    PPPoE interface ATM 1/0.1 will capture remote circuit ID
    PPPoE interface ATM 1/0.1 has 1 active connections,
        out of 1 configured subinterfaces
Assigned profile (any)        : baseProfile
PPPoE Statistics
  Counters:
    PADI received       0
    PADI transmitted    1
    PADO received       1
    PADO transmitted    0
    PADR received       0
    PADR transmitted    1
    PADS received       1
    PADS transmitted    0
    PADT received       0
    PADT transmitted    0
    PADM received       1
    PADM transmitted    0
    PADN received       0
    PADN transmitted    0
  PAD packets received        2
  PAD packets transmitted     2
```

```
  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
    Invalid PAD Tags          0
    Invalid PAD Tag length    0
    Invalid PAD Type          0
    Invalid PADI Session      0
    Invalid PADR Session      0
    Invalid PAD packet length 3
    Invalid PAD packets       0
   Total Invalid PAD packets  3

   Insufficient Resources 0
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

- Example 2—Uses the default MTU value (1494)

```
host1#show pppoe interface full
PPPoE interface FastEthernet 2/0 is operStatusUp
    PPPoE interface FastEthernet 2/0 has max sessions = 8000
    PPPoE interface FastEthernet 2/0 mtu 1494
    PPPoE interface FastEthernet 2/0 has no acName set
    PPPoE interface FastEthernet 2/0 autoconfigured subinterfaces
    PPPoE interface FastEthernet 2/0 has 1 active connections,
        out of 1 configured subinterfaces
Assigned profile (any)        : pppoetest
PPPoE Statistics
  Counters:
    PADI received       42
    PADI transmitted    0
    PADO received       0
    PADO transmitted    8
    PADR received       8
    PADR transmitted    0
    PADS received       0
    PADS transmitted    8
    PADT received       0
    PADT transmitted    7
    PADM received       0
    PADM transmitted    0
    PADN received       0
    PADN transmitted    0
  PAD packets received        50
  PAD packets transmitted     23

  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
    Invalid PAD Tags          0
    Invalid PAD Tag length    0
    Invalid PAD Type          0
    Invalid PADI Session      0
    Invalid PADR Session      0
    Invalid PAD packet length 0
    Invalid PAD packets       0
   Total Invalid PAD packets  0

  Insufficient Resources 0
```

```
Lockout Configuration (seconds): Min 10, Max 120
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

- Example 3—Uses the PPPoE MTU tag

```
host1#show pppoe interface full
PPPoE interface FastEthernet 2/0 is operStatusUp
    PPPoE interface FastEthernet 2/0 has max sessions = 8000
    PPPoE interface FastEthernet 2/0 will use tag value for mtu
    PPPoE interface FastEthernet 2/0 has no acName set
    PPPoE interface FastEthernet 2/0 autoconfigured subinterfaces
    PPPoE interface FastEthernet 2/0 has 1 active connections,
        out of 1 configured subinterfaces
Assigned profile (any)        : pppoetest
PPPoE Statistics
  Counters:
    PADI received       44
    PADI transmitted    0
    PADO received       0
    PADO transmitted    10
    PADR received       10
    PADR transmitted    0
    PADS received       0
    PADS transmitted    10
    PADT received       0
    PADT transmitted    9
    PADM received       0
    PADM transmitted    0
    PADN received       0
    PADN transmitted    0
  PAD packets received        54
  PAD packets transmitted     29

  Invalid PAD Packets:
    Invalid Version           0
    Invalid PAD Code          0
    Invalid PAD Tags          0
    Invalid PAD Tag length    0
    Invalid PAD Type          0
    Invalid PADI Session      0
    Invalid PADR Session      0
    Invalid PAD packet length 0
    Invalid PAD packets       0
   Total Invalid PAD packets  0

  Insufficient Resources 0

Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

show pppoe interface summary

- Use to display the operational and administrative status of all configured PPPoE interfaces.
- Field descriptions
  - Total PPPoE interfaces—Number of configured PPPoE interfaces included in summary
  - Administrative Status
    - Up—Number of interfaces not affected by manual administrative intervention
    - Down—Number of interfaces that cannot flow because of manual administrative intervention
  - Operational Status
    - Up—Number of interfaces that are operational
    - Down—Number of interfaces that are not operational
    - LowerLayerDown—Number of interfaces that are not operational because an underlying interface is down
    - NotPresent—Number of interfaces that are not operational because hardware is unavailable
- Example

```
host1:01#show pppoe interface summary
Total PPPoE interfaces: 16

Administrative Status:
                    Up: 15
                  Down: 1

Operational Status:
                    Up: 15
                  Down: 1
         LowerLayerDown: 1
             NotPresent: 0
```

show pppoe-service-name-table

- Use to display the contents of a PPPoE service name table configured on the router.
- The command displays the table name, action (terminate or drop) associated with the empty service name tag, and the name of each specific service name entry in the table.
- You must specify the name of the PPPoE service name table configured with the **pppoe-service-name-table** command from Global Configuration mode.
- To display the names of PPPoE service name tables that you can specify to complete the command, issue the **show pppoe-service-name-table name ?** command.

- Field descriptions
  - Service Name Table—Name of the PPPoE service name table configured with the **pppoe-service-name-table** command
  - Empty service name action—Action (terminate or drop) associated with the empty service name tag in the PPPoE service name table
  - Service name—Name of the specific (custom) service name tag configured with the **service** command
- Example 1—Displays the names of PPPoE service name tables that you can specify to complete the command

```
host1#show pppoe-service-name-table name ?
  myDefaultTable   myDefaultTable  service-name-table
  myServiceTable1  myServiceTable1 service-name-table
  myServiceTable2  myServiceTable2 service-name-table
  myServiceTable3  myServiceTable3 service-name-table
```

- Example 2—Displays the contents of a default PPPoE service name table with no specific service name entries

```
host1#show pppoe-service-name-table name myDefaultTable
Service Name Table myDefaultTable
 Empty service name action: terminate
```

- Example 3—Displays the contents of a PPPoE service name table that has three specific service name entries and the nondefault action (drop) associated with the empty service name tag

```
host1#show pppoe-service-name-table name myServiceTable1
Service Name Table myServiceTable1
 Empty service name action: drop
 Service name: myISPService
 Service name: myQOSClass1
 Service name: myQOSClass2
```

- Example 4—Displays the names of all PPPoE service name tables configured on the router

```
host1#show pppoe-service-name-table brief
Service-Name Table:
myServiceTable1
myServiceTable2
```

show pppoe subinterface

- Use to display parameters for PPPoE subinterfaces.
- If you do not specify a subinterface, the router displays the configured PPPoE subinterface number and status for all configured PPPoE subinterfaces.
- If you specify an interface with no subinterface, the router displays the status for the subinterfaces associated with the interface.
- If you specify an interface and subinterface, the router displays detailed parameters available for that subinterface.
- To display configuration, status, and statistics information, use the **full** keyword.

- Field descriptions
  - PPPoE subinterface—Interface specifier
  - Status—Operational status of the interface. Possible values are:
    - operStatusUp—Interface or subinterface is operational
    - Down—Interface or subinterface is not operational
    - LowerLayerDown—Subinterface is not operational because an underlying interface or subinterface is down
  - URL String—URL string sent in the PADM message to PPPoE clients
  - MOTM String—Message of the minute string sent in the PADM message to PPPoE clients
  - session id—Session ID of the subinterface
  - source MAC address—MAC address of PPPoE client
  - MTU—Maximum transfer unit (MTU) value; when derived from the PPPoE MTU tag, the value can only be determined from an active session.
  - In Octets—Number of octets received on the subinterface
  - Out Octets—Number of octets transmitted on the subinterface
  - In Packets—Number of packets received on the subinterface
  - Out Packets—Number of packets transmitted on the subinterface
- Example 1

```
host1:v0#show pppoe subinterface fastEthernet 1/1.1.1
PPPoE subinterface fastEthernet 1/1.1.1 is operStatusUp
      URL String: http://www.urlofinterest.com
      MOTM String: a horse walks into a bar
PPPoE subinterface fastEthernet 1/1.1.1 has a session id of 1
PPPoE Statistics
        In Octets:  480
        Out Octets: 256
        In Packets:  8
        Out Packets: 8
```

- Example 2

```
host1:v0#show pppoe subinterface full
PPPoE subinterface FastEthernet 2/0.11 is operStatusUp (dynamic)
    PPPoE subinterface FastEthernet 2/0.11 has a session id of 8
    PPPoE subinterface FastEthernet 2/0.11 has source MAC address 0090.1a40.280a
    PPPoE subinterface FastEthernet 2/0.11 has a MTU of 1494
PPPoE Statistics
    In Octets:  165922
    Out Octets: 108283
    In Packets:  3607
    Out Packets: 3608
```

show pppoe subinterface summary

- Use to display the operational and administrative status of all configured PPPoE subinterfaces.
- Field descriptions
  - Total PPPoE subinterfaces—Number of configured PPPoE subinterfaces included in summary
  - Administrative Status
    - Up—Number of subinterfaces not affected by manual administrative intervention
    - Down—Number of subinterfaces that cannot flow because of manual administrative intervention
  - Operational Status
    - Up—Number of subinterfaces that are operational
    - Down—Number of subinterfaces that are not operational
    - LowerLayerDown—Number of subinterfaces that are not operational because an underlying interface is down
    - NotPresent—Number of subinterfaces that are not operational because hardware is unavailable
- Example

```
host1:01#show pppoe subinterface summary
Total PPPoE subinterfaces: 116

Administrative Status:
                    Up: 115
                  Down: 1

Operational Status:
                    Up: 115
                  Down: 1
         LowerLayerDown: 1
             NotPresent: 0
```

show profile

- Use to display information about profiles.
- To display information about a specific profile, use the **name** keyword.
- To display a list of profiles configured on the router, use the **brief** keyword.
- Field descriptions
  - Profile—Name of the profile that is displayed
  - 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—Name of the virtual router (VR) assigned to the profile; interfaces created by the profile are attached to this VR
  - Directed Broadcast—Enabled or disabled

- ICMP Redirects—Enabled or disabled
- Access Route Addition—Enabled or disabled
- Network Address Translation—Enabled or disabled; domain location (inside or outside)
- Source-Address Validation—Enabled or disabled
- Ignore DF Bit—Enabled or disabled
- Filter Option Packets—Router filters out packets with IP options; enabled or disabled
- Administrative MTU—MTU size configured on the profile
- TCP MSS value—Maximum segment size for TCP SYN packets traveling through the interface
- Inactivity Timer—Inactivity timer setting; enabled or disabled
- Route Map Name—Route map applied to the IP interface subscriber; enabled or disabled
- 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
- IGMP—Enabled or disabled
- static-groups—Displays address of any static groups configured for IGMP
- Input policy—Name of input policy and whether statistics are enabled or disabled
- Output policy—Name of output policy and whether statistics are enabled or disabled
- PPP Keepalive—PPP keepalive period, in seconds
- PPP Magic Number—Enabled or disabled
- PPP Peer DNS Priority—Enabled or disabled
- PPP Peer WINS Priority—Enabled or disabled
- PPP Authentication—Type of authentication configured: PAP, CHAP, or none
- PPP Authentication Router—Name of authentication virtual router
- PPP Negotiate MRU—MRU configured for the profile
- PPP Packet Log—Enabled or disabled
- PPP State Log—Enabled or disabled
- PPP Chap Challenge Length—Minimum and maximum Chap Challenge length
- PPP Passive Mode—Enabled or disabled
- PPP Multilink—Enabled or disabled
- PPP IPCP netmask option—Enabled or disabled
- PPP AAA Profile—AAA profile associated with this PPP interface

- PPP Multilink Fragmentation—Enabled or disabled
- PPP Multilink Fragment Size—Multilink fragment size for this PPP interface
- PPP Multilink Reassembly—Enabled or disabled
- PPP Multilink Mrru—Multilink MRRU value for this PPP interface
- PPP Initiate IP—Initiation of IPv4 over this PPP interface; enabled or disabled
- PPP Initiate IPv6—Initiation of IPv6 over this PPP interface; enabled or disabled
- PPPoE Max Sessions—Maximum number of PPPoE subinterfaces that can be on an interface
- PPPoE Always-offer—Router offers to set up a session for the client, even if the router has insufficient resources to establish a session; enabled or disabled
- PPPoE Remote-Circuit-Id—The router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM); enabled or disabled
- PPPoE Log PPPoeControlPacket—Enabled or disabled
- PPPOE duplicate-protect—Enabled or disabled
- PPPoE ACNAME—Access concentrator name
- PPPoE URL—URL sent in PADM message to PPPoE clients
- PPPoE MOTM—Message of the minute sent in the PADM message to PPPoE clients
- PPPoE Service-Name Table—Name of the PPPoE service name table, if configured for the specified profile

- Example—Displays configuration information for a PPPoE profile assigned to a dynamic interface

```
host1#show profile name pppoeProfile
Profile                    : pppoeProfile
Unnumbered interface on    : loopback 1
Router                     : default
Directed Broadcast         : Disabled
ICMP Redirects             : Disabled
Access Route Addition      : Enabled
Network Address Translation: Disabled
Source-Address Validation  : Disabled
Ignore DF Bit              : Disabled
Filter Option Packets      : Disabled
Administrative MTU         : 1500
TCP MSS value              : 0
Inactivity Timer           : Disabled
Route Map Name             : Disabled
Auto Detect                : Disabled
Auto Configure             : Disabled

IGMP                       : Enabled
  static-groups            :
  Input policy:  bobb statistics enabled
  Output policy: bobb statistics enabled
```

```
PPP Keepalive               : 30
PPP Magic Number            : enabled
PPP Peer DNS Priority       : disabled
PPP Peer WINS Priority      : disabled
PPP Authentication          : pap/chap
PPP Authentication Router   :
PPP Negotiate MRU           : (use lower layer MRU)
PPP Packet Log              : disabled
PPP State Log               : disabled
PPP Chap Challenge Length   : 16 - 32
PPP Passive Mode            : disabled
PPP Multilink               : disabled
PPP IPCP Netmask Option     : disabled
PPP AAA Profile             :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly    : disabled
PPP Multilink Mrru          : (use MRU)
PPP Initiate IP                   : disabled
PPP Initiate IPv6                 : disabled
PPPoE Max Sessions          : 2
PPPoE Always-offer          : Disabled
PPPoE Remote-Circuit-Id     : Enabled
PPPoE Log PPPoeControlPacket: Disabled
PPPoE duplicate-protect     : Enabled
PPPoE ACNAME                : CYM9876
PPPoE URL                   : http://www.urlofinterest.com
PPPoE MOTM                  : goodmorning
PPPoE Service-Name table    : myServiceTable1
```

show radius override

- Use to display the current override settings configured on the RADIUS client (LNS) for the NAS-IP-Address [4], NAS-Port-Id [87], Calling-Station-Id [31], and NAS-Identifier [32] RADIUS attributes.
- Field descriptions
  - nas-ip-addr—Override setting for the NAS-IP-Address attribute
  - nas-port-id—Override setting for the NAS-Port-Id attribute; value is remote-circuit-id if configured with **radius override nas-port-id remote-circuit-id** command
  - calling-station-id—Override setting for the Calling-Station-Id attribute; value is remote-circuit-id if configured with **radius override calling-station-id remote-circuit-id** command
  - nas-info—Virtual router that generates the NAS-IP-Address and NAS-Identifier attributes for AAA broadcast accounting packets; current virtual router or authentication virtual router
- Example

```
host1#show radius override
nas-ip-addr:        nas-ip-addr
nas-port-id:        remote-circuit-id
calling-station-id: remote-circuit-id
nas-info:           from current virtual router
```

show radius remote-circuit-id-delimiter

- Use to display the delimiter character configured to set off components in the PPPoE remote circuit ID value captured from a DSLAM.
- The default delimiter character is #.
- Example

```
host1#show radius remote-circuit-id-delimiter
%
```

show radius remote-circuit-id-format

- Use to display the format configured for the PPPoE remote circuit ID value captured from a DSLAM.
- If the PPPoE remote circuit ID value is configured to include any or all of the agent-circuit-id, agent-remote-id, and nas-identifier components, the display lists the components included and the order in which they appear.
- If the PPPoE remote circuit ID value is configured to use the format for the **dsl-forum-1** keyword of the **radius remote-circuit-id-format** command, the display indicates that this format is in effect.
- The default format is agent-circuit-ID.
- Example

```
host1#show radius remote-circuit-id-format
nas-identifier agent-circuit-id agent-remote-id
```

Troubleshooting

Use the **pppoeControlPacket** log to diagnose problems on your PPPoE interfaces.

log severity debug pppoeControlPacket

- Use to configure a packet trace log for a PPPoE interface. You must specify a PPPoE major interface.
- Specify one of the following interface types and a corresponding interface specifier. For more information, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - **fastEthernet**
  - **gigabitEthernet**
  - **atm**
  - **tenGigabitEthernet**
- The packet trace log for a PPPoE interface displays only the first 256 bytes of packet data. Data in excess of 256 bytes does not appear in the packet trace log.
- You also configure logging to direct the output to a specific destination. For information, see *JUNOSe System Event Logging Reference Guide, Chapter 1, System Logging Overview*.

- Example

```
host1(config-if)#ip address 164.10.6.71 255.255.255.0
host1(config-if)#log severity debug pppoeControlPacket atm 10/0.1
host1:v0#DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADI rx from
00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADO tx to 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADR rx from 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:19 pppoeControlPacket (interface atm 10/0.1): PADS tx to 00-09-01-a0-00-2e,
connection made  using session id 3 on sub interface 1

RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#
RX-a0-00-2e:v0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
RX-a0-00-2e:v0(config)#interface atm 10/1.1.1
RX-a0-00-2e:v0(config-if)#ppp shut
RX-a0-00-2e:v0(config-if)#DEBUG 07/25/2000 15:13:38 pppoeControlPacket (interface atm 10/0.1): PADT rx
from 00-09-01-a0-00-2e
RX-a0-00-2e:v0(config-if)#
RX-a0-00-2e:v0(config-if)#no ppp shut
RX-a0-00-2e:v0(config-if)#pppoe test
RX-a0-00-2e:v0(config-if)#DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADI rx
from 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADO tx to 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADR rx from 00-09-01-a0-00-2e
DEBUG 07/25/2000 15:13:49 pppoeControlPacket (interface atm 10/0.1): PADS tx to 00-09-01-a0-00-2e,
connection made  using session id 4 on sub interface 1

RX-a0-00-2e:v0(config-if)#
RX-a0-00-2e:v0(config-if)#exit
```

Chapter 11

Configuring Bridged IP

E-series routers support bridged IP (1483) interfaces.

This chapter contains the following sections:

Overview

You can configure a bridged IP interface to manage IP packets that are encapsulated inside an Ethernet frame running over a permanent virtual circuit (PVC).

When you configure a bridged IP interface, it automatically performs proxy Address Resolution Protocol (ARP). You can also configure the router as a relay agent that forwards Dynamic Host Configuration Protocol (DHCP) broadcasts.

Proxy ARP

Proxy ARP allows your router to respond to ARP requests on behalf of an Ethernet end node.

The router performs proxy ARP for the ARP requests that come in over the bridged IP interface when both of the following conditions are met:

- The IP address in the ARP request matches an entry in the routing table.
- The route is on a different interface than the one on which the router received the ARP request.

If you specify that the bridged IP interface performs unrestricted proxy ARP, it also performs proxy ARP when the route is on the interface that received the ARP request.

In most situations, do not configure the router to perform unrestricted proxy ARP. Do so for special situations, such as when cable modems are used. When an IP client broadcasts the ARP request across the Ethernet wire, the end node with the correct IP address responds to the ARP request and provides the correct MAC address. If the unrestricted proxy ARP feature is enabled, the router response is redundant and might fool the IP client into determining that the destination MAC address within its own subnet is the same as the address of the router.

DHCP

DHCP provides a mechanism through which hosts using TCP/IP can obtain protocol configuration parameters automatically from a DHCP server on the network.

The most important configuration parameter carried by DHCP is the IP address. A host must be initially assigned a specific IP address that is appropriate to the network to which the computer is attached, and that is not assigned to any other host on that network. If you move a host to a new network, you must give it a new IP address.

DHCP also carries other important configuration parameters such as the subnet mask, default router, and Domain Name System (DNS) server.

An IP client contacts a DHCP server for configuration parameters. The DHCP server is typically centrally located and operated by the network administrator. Because a network administrator manages the server, DHCP clients can obtain reliable parameters appropriate to the current network architecture.

For information about DHCP, see *JUNOSe Broadband Access Configuration Guide, Chapter 17, DHCP Overview*.

Platform Considerations

You can configure bridged IP interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support bridged IP interfaces on ERX-14xx models, ERX-7xx 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 bridged IP.

For information about the modules that support bridged IP interfaces 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 bridged IP.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the ATM physical interface on which you want to configure bridged IP. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about bridged IP, consult RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999). Note that RFC 2684 obsoletes RFC 1483.

Before You Configure Bridged IP

Before you configure bridged IP on an ATM interface, verify that:

- You have correctly installed a module that supports bridged IP. For information, see *ERX Module Guide, Appendix A, Module Protocol Support* (on ERX-7xx models, ERX-14xx models, and the ERX-310 router) or *E120 and E320 Module Guide*, *Appendix A*, *IOA Protocol Support* (on the E120 router or the E320 router).
- Each configured line can transmit data to and receive data from your switch connections.

Table 20 lists the prerequisite tasks for configuring bridged IP and the resources that you can consult to learn how to perform these tasks.

Table 20: Prerequisite Tasks for Configuring Bridged IP

To Learn About	See
Preconfiguration and hardware diagnostic procedures	*ERX Hardware Guide* *E120 and E320 Hardware Guide*
Configuring T3 and E3 line modules	*JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*
Configuring OC3 line modules	*JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*

Also have the following information available:

- Interface specifiers for the ATM interfaces on which you want to configure bridged IP

  For more information about specifying ATM interfaces and subinterfaces on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
- Subinterface numbers for each logical interface that you want to create
- Virtual path and channel numbers for each virtual circuit that you want to create
- IP addresses and subnet mask assignments for IP interfaces
- IP address of the DHCP server

Configuring Bridged IP

To configure an ATM interface using bridged IP encapsulation:

1. Configure a physical interface.

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

2. Configure the subinterface.

   host1(config-if)#**interface atm 0/1.20**

3. Configure a PVC on the subinterface by specifying the virtual circuit descriptor (VCD), the virtual path identifier (VPI), the virtual channel identifier (VCI), and the encapsulation type.

   host1(config-if)#**atm pvc 10 22 100 aal5snap**

4. Configure bridged IP encapsulation.

   host1(config-if)#**encapsulation bridge1483**

5. Assign an IP address and subnet mask to the PVC.

   host1(config-subif)#**ip address 192.168.10.20 255.255.255.0**

NOTE: You can also assign an IP template to the interface or create an unnumbered interface instead of assigning an IP address. For details, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*.

6. (Optional) Use the appropriate **show** commands to verify your configuration.

   host1#**show atm interface 0/1**
   host1#**show atm vc 0/1 10**
   host1#**show atm subinterface 0/1.20**

   For more information about using these commands, see *Monitoring ATM* on page 66 in *Chapter 1, Configuring ATM*.

atm pvc

- Use to configure a PVC on an ATM interface.
- The following fields are mandatory:
  - *vcd*—Unique number that identifies a virtual circuit in the range 1–2147483647. The VCD value has no relationship to the VPI and VCI values and has meaning only to the E-series router.
  - *vpi*—8-bit field in the ATM cell header. The VPI value is unique on a single link, not throughout the ATM network, because it has meaning only to the E-series router. The VPI value must match the value on the ATM switch.

NOTE: Do not set both the VPI and VCI values to zero.

- *vci*—16-bit field in the ATM cell header. The VCI value is unique on a single link, not throughout the ATM network, because it has meaning only to the router. You cannot set both the VPI and VCI to 0.
- *encapsulation type*:
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit. An LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
  - **aal5muxip**—Specifies a multiplexed circuit used for IP only.

- Example

  host1(config-if)#**atm pvc 10 22 100 aal5snap**

- Use the **no** version to remove the specified PVC.

encapsulation bridge1483

- Use to configure bridged IP as the encapsulation method on an interface.
- Use the **unrestrictedProxyArp** keyword to allow the router to perform unrestricted processing of ARP requests even if the route is on the same interface on which the request is received. See *Proxy ARP* on page 363 for details.
- Example

  host1(config-if)#**encapsulation bridge1483**

- Use the **no** version to remove bridged IP as the encapsulation method on the interface.

interface atm

- Use to configure an ATM interface.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647

- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config)#**interface atm 3/1.20**
  host1(config)#**interface atm 5/0/1.20**

- Use the **no** version to remove the ATM subinterface or the logical interface.

Chapter 12

Configuring Bridged Ethernet

This chapter describes how to configure bridged Ethernet on E-series routers.

E-series routers also support bridged Ethernet on dynamic interfaces. See *Chapter 15, Configuring Dynamic Interfaces*, for details.

This chapter contains the following sections:

Overview

Bridged Ethernet allows multiple upper-layer interface types (IP and PPPoE) to be simultaneously multiplexed over the same interface. You can set up the router to either terminate interfaces and route data or to pass data transparently through the router to another terminating device. This capability supports multiple client devices that use both IP and Point-to-Point Protocol over Ethernet (PPPoE) encapsulation over an Ethernet LAN.

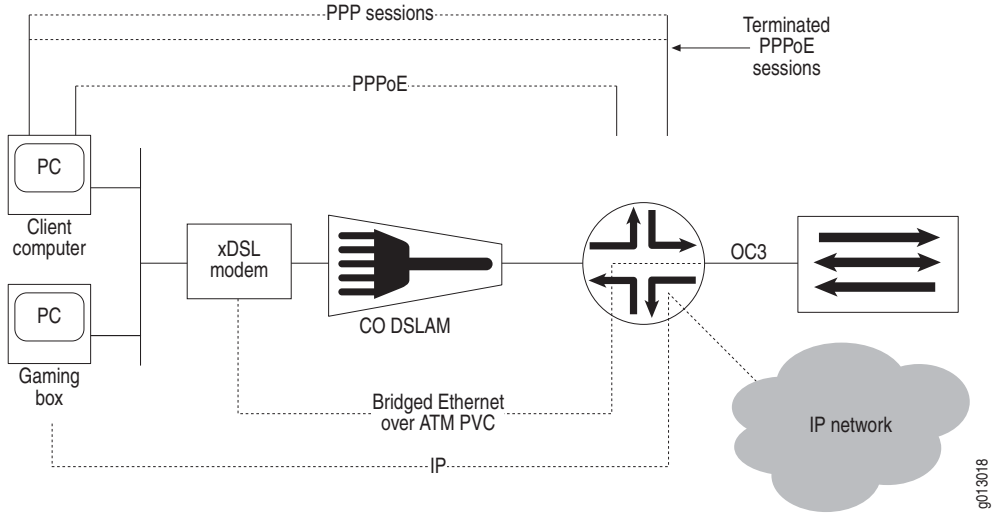
NOTE: Although connection-based forwarding is *not* supported on any E-series router, as an alternative, you can configure a local cross-connect, which uses layer 2 services over MPLS to transmit data between two layer 2 interfaces that reside on the same E-series router. Configuration of local cross-connects is supported on all E-series routers. For more information about configuring local cross-connects, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.

Bridged Ethernet Application

Figure 36 shows an example of a client computer using IP/PPP/PPPoE and an Internet gaming system running IP, connecting to the E-series router over the same ATM PVC. The client computer and gaming system can connect to an E-series router via an xDSL modem over a single ATM PVC, and the router can forward these two data streams independently. When the router receives the two data streams, it uses the Ethertype contained in the bridged Ethernet header to select which upper interface (IP or PPPoE) receives the frame.

In Figure 36, IP and PPPoE interfaces are configured so that the non-PPPoE IP traffic is received by the IP interface, and the IP/PPP/PPPoE traffic is received by the PPPoE interface. Since the router receives these data streams on different IP interfaces, they may be routed independently.

Figure 36: Bridged Ethernet Topology, Router Terminating and Routing Traffic



Assigning MAC Addresses

When you create a bridged Ethernet interface, the system media access control (MAC) address is assigned to the interface by default. However, you can assign a specific MAC address to each statically configured bridged Ethernet interface. For example, if multiple statically configured bridged Ethernet interfaces are connected to the same device, using specific MAC addresses enables the connected device to select the correct ATM port or VC to use.

You configure a specific MAC address when you create the bridged Ethernet interface. If you want to modify an existing MAC address, you must remove the interface and create it again. Also, you cannot configure multicast MAC addresses on bridged Ethernet interfaces.

VLAN and S-VLAN Configurations

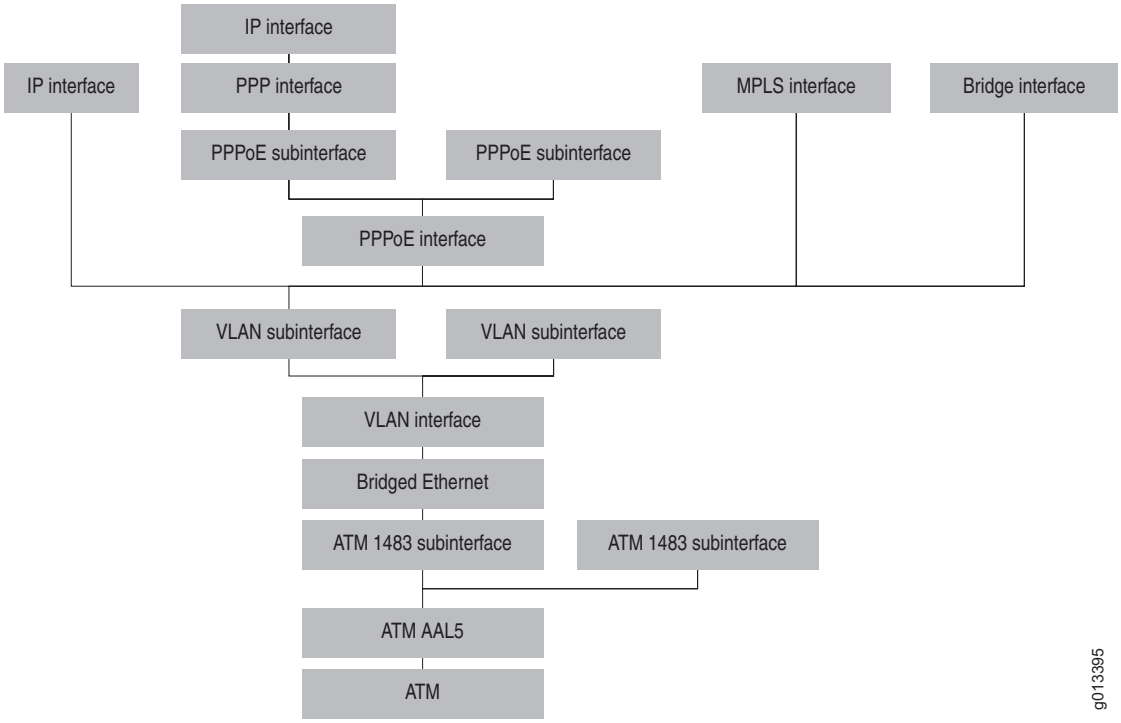
Bridged Ethernet interfaces on E-series routers support the configuration of virtual local area networks (VLANs) and stacked virtual local area networks (S-VLANs). A VLAN permits multiplexing multiple higher-level protocols over a single physical port. An S-VLAN provides a two-level VLAN tag structure, which extends the VLAN ID space to more than 16 million VLAN tags.

Specifically, you can statically configure the following higher-level protocols over a VLAN or an S-VLAN subinterface that is stacked above a bridged Ethernet interface:

- IP
- MPLS
- PPPoE
- Transparent bridging

Figure 37 illustrates the interface stacking supported on E-series routers for VLANs over bridged Ethernet.

Figure 37: Interface Stacking for VLANs over Bridged Ethernet



VLANs and S-VLANs configured over bridged Ethernet interfaces provide the same basic capabilities as VLANs and S-VLANs configured over Ethernet interfaces, with the following exception:

- S-VLAN oversubscription is not supported on bridged Ethernet interfaces.

After you configure the bridged Ethernet interface, you configure the VLANs, S-VLANs, and the supported higher-level protocols in the same way that you configure them over Ethernet interfaces.

For more information, see:

- *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces* for introductory information about VLANs and S-VLANs.
- *Configuring VLANs over Bridged Ethernet* on page 381 and *Configuring S-VLANs over Bridged Ethernet* on page 385 for examples that illustrate VLAN and S-VLAN configurations over bridged Ethernet.

Platform Considerations

You can configure bridged Ethernet on the following E-series routers:

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

Module Requirements

For information about the modules that support bridged Ethernet on ERX-14xx models, ERX-7xx 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 bridged Ethernet.

For information about the modules that support bridged Ethernet 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 bridged Ethernet.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the ATM physical interface on which you want to configure bridged Ethernet. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about bridged Ethernet, consult the following resources:

- RFC 826—An Ethernet Address Resolution Protocol (November 1982)
- RFC 2684—Multiprotocol Encapsulation over ATM Adaptation Layer 5 (September 1999)

Configuring Bridged Ethernet

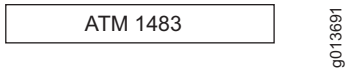
This section shows how to configure IP with PPPoE terminated at the E-series router. With each step, an illustration shows how the router is building the interface columns.

Configuring IP with PPPoE Terminated at the Router

This section shows how to create IP with PPPoE interfaces that terminate the connection and route the data received on the PVC, as shown in the example in Figure 36 on page 372. To create a terminated PVC:

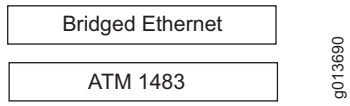
1. Create an ATM 1483 subinterface and associated PVC.

   host1(config)#**interface atm 9/1.1 point-to-point**
   host1(config-subif)#**atm pvc 1 0 32 aal5snap 0 0 0**

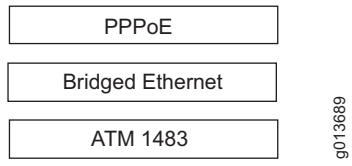
2. Encapsulate the ATM 1483 subinterface with bridged Ethernet. The use of the **encapsulation** keyword implies that the bridged Ethernet interface is the only interface stacked directly above the ATM 1483 subinterface. As a result, the bridged Ethernet interface cannot have a peer interface stacked above the same lower-layer interface.

   host1(config-subif)#**encapsulation bridge1483**

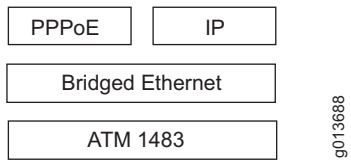
3. Create a PPPoE major interface over the bridged Ethernet interface. This command does not use the **encapsulation** keyword.

   host1(config-subif)#**pppoe**

4. Create an IP interface over the bridged Ethernet interface as a peer to the PPPoE interface.

   host1(config-subif)#**ip address 160.1.0.1 255.255.255.0**

5. (Optional) Set up the router to validate MAC addresses on the IP interface.

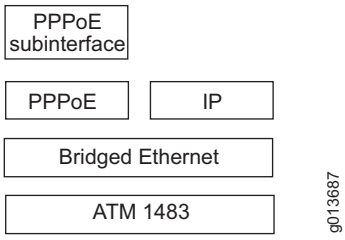
   host1(config-subif)#**ip mac-validate strict**

6. Exit the subinterface context.

   host1(config-subif)#**exit**

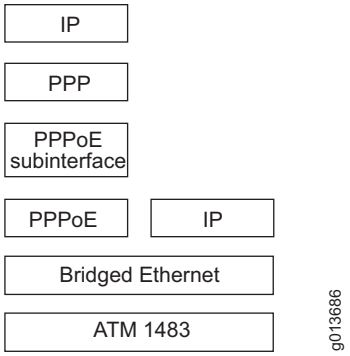
7. Create a PPPoE subinterface over the major interface.

   host1(config)#**pppoe subinterface atm 9/1.1.1**



8. Configure PPP encapsulation over the PPPoE subinterface, and the IP interface over the PPP interface.

   host1(config-subif)#**encapsulation ppp**
   host1(config-subif)#**ip address 160.1.1.1 255.255.255.0**



atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 1 0 32 aal5snap 0 0 0**

- Use the **no** version to remove the specified PVC.

encapsulation bridge1483

- Use to configure bridged Ethernet as the encapsulation method on an interface, and to optionally assign the MAC address to the interface.
- Use the **mac-address** keyword to configure a specific MAC address for the interface. Otherwise, the system MAC address is used. The same MAC address can be used on multiple interfaces.
- If the MAC address is configured, you must use the same MAC address whenever you reenter the **encapsulation bridge1483** command for the interface.
- The MAC address can be configured only when the interface is created. To change a MAC address, you must remove the interface and create it again.
- Example

  host1(config-subif)#**encapsulation bridge1483 mac-address 0090.1a01.1234**

- Use the **no** version, without the MAC address, to remove bridged Ethernet as the encapsulation method on the interface.

encapsulation ppp

- Use to configure PPP as the encapsulation method for an interface.
- Example

  host1(config-subif)#**encapsulation ppp**

- Use the **no** version to remove PPP as the encapsulation method on the interface.

interface atm

- Use to configure an ATM interface or subinterface type.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647

- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config)#**interface atm 9/1.1 point-to-point**
  host1(config)#**interface atm 5/0/1.1 point-to-point**

- Use the **no** version to remove the interface or subinterface.

ip address

- Use to set an IP address for the interface.
- Note that you cannot add more than one IP address to bridged Ethernet interfaces.
- Example

  host1(config-subif)#**ip address 160.1.0.1 255.255.255.0**

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

ip mac-validate

- Use to enable or disable MAC address validation on a per interface basis.
- When MAC address validation is enabled, the router checks the entry in the MAC validation table that corresponds to the IP source address of an incoming packet. The MAC source address of the packet must match the MAC source address of the table entry for the router to forward the packet.
- Use the **strict** keyword to prevent transmission of IP packets that do not reside in the validation table.
- Use the **loose** keyword to allow IP packets to pass through even though the packets do not have entries in the validation table. Only packets that have matching IP–MAC pair entries in the table are validated.
- The default behavior is not to perform MAC address validation.
- Example

  host1(config-subif)#**ip mac-validate strict**

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

NOTE: For more information, see *MAC Address Validation* in *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*.

pppoe

- Use to create a PPPoE major interface.
- Example

  host1(config-subif)#**pppoe**

- Use the **no** version to remove the PPPoE major interface.

pppoe subinterface atm

- Use to create a PPPoE subinterface on an ATM interface.
- On ERX-7xx models, ERX-14xx models, and the ERX-310 router, use the *slot/port.atmSubinterface.pppoeSubinterface* format.
- On the E120 and E320 routers, use the *slot/adapter/port.atmSubinterface.pppoeSubinterface* format.
- Examples

  host1(config)#**pppoe subinterface atm 9/1.1.1**
  host1(config)#**pppoe subinterface atm 5/0/1.1.1**

- Use the **no** version to remove the PPPoE subinterface.

Alternative Configuration

In previous releases, you could configure a PPPoE major interface directly over ATM 1483 only. The router still supports this stacking and configuration method for PPPoE. Although the older and newer interface stacks are different, they behave the same in terms of frame encapsulation and handling. As a result, an interface created using the older stacking will interoperate with an interface using the new stacking. Note, however, that the previous command syntax (**encapsulation pppoe**) cannot be used when a bridged Ethernet interface already exists, because it is intended to produce the old stacking for PPPoE over ATM 1483.

1. Create the ATM 1483 subinterface and associated PVC:

   host1(config)#**interface atm 9/1.1 point-to-point**
   host1(config-subif)#**atm pvc 1 0 32 aal5snap 0 0 0**

2. Create a PPPoE major interface over the ATM 1483 subinterface. Note that since this command uses the **encapsulation** keyword, it will fail if a bridged Ethernet interface was already created over the ATM 1483 subinterface using the new syntax.

   host1(config-subif)#**encapsulation pppoe**

3. Create a PPPoE subinterface over the major interface. Because PPPoE is the only top layer protocol in the stack, there is no need to use **pppoe** to identify the subinterface type (it is implied).

   host1(config)#**interface atm 9/1.1.1**

4. Configure the PPP encapsulation over the PPPoE subinterface, and the IP interface over the PPP interface.

   host1(config-subif)#**encapsulation ppp**
   host1(config-subif)#**ip address 160.1.1.1 255.255.255.0**

Configuring VLANs over Bridged Ethernet

This section describes how to create the following common static VLAN over bridged Ethernet configurations:

- IP over VLAN over bridged Ethernet
- PPPoE over VLAN over bridged Ethernet
- MPLS over VLAN over bridged Ethernet

You can also configure transparent bridging over VLANs over bridged Ethernet. For information about configuring transparent bridging, see *Chapter 13, Configuring Transparent Bridging*.

Configuring VLANs over bridged Ethernet interfaces consists of two basic tasks:

1. Configure the layers up to and including the VLAN subinterface. The steps for this task are common to all configurations.
2. Configure the higher-level protocols above the VLAN subinterface.

The following sections describe how to configure VLANs over bridged Ethernet. For more information about the commands used in these procedures, see the command descriptions listed in alphabetical order at the end of *Configuring Higher-Level Protocols over VLANs* on page 382.

NOTE: Before you can remove a VLAN subinterface, you must remove the upper-layer interface stack.

NOTE: For more information about specifying ATM interfaces and subinterfaces, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Configuring VLAN Subinterfaces over Bridged Ethernet

To configure a VLAN subinterface over bridged Ethernet:

1. Create an ATM 1483 subinterface and associated PVC.

   host1(config)#**interface atm 4/0.101**
   host1(config-subif)#**atm pvc 1 0 32 aal5snap 0 0 0**

2. Specify bridged Ethernet as the encapsulation method for the ATM 1483 subinterface.

   host1(config-subif)#**encapsulation bridge1483**

3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the bridged Ethernet interface.

   host1(config-subif)#**encapsulation vlan**

4. Create a VLAN subinterface to carry the higher-level protocols by adding a subinterface number to the interface identification command.

   host1(config-subif)#**interface atm 4/0.101.1**

5. Assign a VLAN ID for the subinterface.

   host1(config-subif)#**vlan id 10**

6. (Optional) Configure additional VLAN subinterfaces by repeating Steps 4 and 5, using unique values.

   host1(config-subif)#**interface atm 4/0.101.2**
   host1(config-subif)#**vlan id 11**

Proceed to the next section for instructions on configuring higher-level protocols over the VLAN subinterfaces.

Configuring Higher-Level Protocols over VLANs

This section provides examples for configuring IP, PPPoE, and MPLS interfaces over VLAN subinterfaces configured on bridged Ethernet. These procedures assume that you have already configured one or more VLAN subinterfaces over the bridged Ethernet interface to carry the higher-level protocols.

Configuring IP over VLAN

To configure IP over VLAN over a bridged Ethernet interface:

1. Follow the steps in *Configuring VLAN Subinterfaces over Bridged Ethernet* on page 381 to configure the VLAN subinterface.

2. Assign an IP address and mask to the VLAN subinterface.

   host1(config-subif)#**ip address 10.1.1.1 255.255.255.0**

Configuring PPPoE over VLAN

To configure PPPoE over VLAN over a bridged Ethernet interface:

1. Follow the steps in *Configuring VLAN Subinterfaces over Bridged Ethernet* on page 381 to configure the VLAN subinterface.

2. Create a PPPoE major interface on the VLAN subinterface.

   host1(config-subif)#**pppoe**

3. Exit the subinterface context.

   host1(config-subif)#**exit**

4. Create a PPPoE subinterface by adding a subinterface number to the interface identification command.

   host1(config)#**pppoe subinterface atm 4/0.101.2.1**

5. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**

6. Assign an IP address and mask to the interface.

   host1(config-subif)#**ip address 10.1.1.2 255.255.255.0**

Configuring MPLS over VLAN

To configure MPLS over VLAN over a bridged Ethernet interface:

1. Follow the steps in *Configuring VLAN Subinterfaces over Bridged Ethernet* on page 381 to configure the VLAN subinterface.

2. Enable MPLS on the VLAN subinterface.

   host1(config-subif)#**mpls**

atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 1 5 50 aal5snap 0 0 0**

- Use the **no** version to remove the specified PVC.

encapsulation bridge1483

- Use to configure bridged Ethernet as the encapsulation method on an ATM 1483 subinterface.
- Example

  host1(config-subif)#**encapsulation bridge1483**

- Use the **no** version to remove bridged Ethernet as the encapsulation method on the interface.

encapsulation ppp

- Use to configure PPP as the encapsulation method on an interface.
- Example

  host1(config-subif)#**encapsulation ppp**

- Use the **no** version to remove PPP as the encapsulation method on the interface.

encapsulation vlan

- Use to configure VLAN as the encapsulation method on an interface.
- Example

  host1(config-subif)#**encapsulation vlan**

- Use the **no** version to remove VLAN as the encapsulation method on the interface.

interface atm

- Use to configure an ATM interface, ATM 1483 subinterface, or VLAN subinterface.
- On ERX-7xx models, ERX-14xx models, and the ERX-310 router, use the *slot/port.subinterface.vlanSubinterface* format.
- On E120 and E320 routers, use the *slot/adapter/port.subinterface.vlanSubinterface* format.
- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Example 1—Configures a VLAN subinterface over bridged Ethernet on ERX-7xx models, ERX-14xx models, and the ERX-310 router

  host1(config)#**interface atm 4/2.2 point-to-point**
  host1(config-subif)#**interface atm 4/2.2.3**

- Example 2—Configures a VLAN subinterface over bridged Ethernet on the E320 router

  host1(config)#**interface atm 4/0/2.2 point-to-point**
  host1(config-subif)#**interface atm 4/0/2.2.3**

- Use the **no** version to remove the interface or subinterface.

ip address

- Use to set an IP address for the interface.
- Note that you cannot add more than one IP address to bridged Ethernet interfaces.
- Example

  host1(config-subif)#**ip address 10.1.2.3 255.255.255.255**

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

mpls

- Use to enable, disable, or delete MPLS on an interface. MPLS is disabled by default.
- Example

  host1(config)#**mpls**

- Use the **no** version to halt MPLS on the interface and delete the MPLS interface configuration.

pppoe

- Use to create a PPPoE major interface.
- Example

  host1(config-subif)#**pppoe**

- Use the **no** version to remove the PPPoE major interface.

pppoe subinterface atm

- Use to create a PPPoE subinterface over a VLAN subinterface configured on a bridged Ethernet interface.
- On ERX-7xx models, ERX-14xx models, and the ERX-310 router, use the *slot/port.atmSubinterface.vlanSubinterface.pppoeSubinterface* format.
- On E120 and E320 routers, use the *slot/adapter/port.atmSubinterface.vlanSubinterface.pppoeSubinterface* format.
- Examples

  host1(config)#**pppoe subinterface atm 4/0.1.2.1**
  host1(config)#**pppoe subinterface atm 4/1/0.1.2.1**

- Use the **no** version to remove the PPPoE subinterface.

vlan id

- Use to specify the VLAN ID.
- Use a VLAN ID that is in the range 0–4095 and is unique within the interface.
- Issue the **vlan id** command before any upper bindings are made, such as IP or PPPoE.
- Use the optional keyword **untagged** to specify that frames be sent untagged. The keyword is valid only for VLAN ID 0. It allows tagged frames to be received, but sends out untagged frames.
- Example

  host1(config-subif)#**vlan id 400**

- There is no **no** version.

Configuring S-VLANs over Bridged Ethernet

S-VLANs over bridged Ethernet support the same set of higher-level protocols as VLANs over bridged Ethernet. You configure S-VLANs over bridged Ethernet in the same way that you configure VLANs over bridged Ethernet, with the addition of certain commands.

Like VLANs, configuring S-VLANs over bridged Ethernet interfaces consists of two basic tasks:

1. Configure the layers up to and including the S-VLAN subinterface.
2. Configure the higher-level protocols above the S-VLAN subinterface.

Before you can remove an S-VLAN subinterface, you must remove the upper-layer interface stack.

NOTE: S-VLAN oversubscription is not supported on bridged Ethernet interfaces.

NOTE: For more information about specifying ATM interfaces and subinterfaces, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Configuring S-VLAN Subinterfaces over Bridged Ethernet

To configure an S-VLAN subinterface over bridged Ethernet:

1. Create an ATM 1483 subinterface and associated PVC.

   host1(config)#**interface atm 3/1.1**
   host1(config-subif)#**atm pvc 1 5 33 aal5snap 0 0 0**

2. Specify bridged Ethernet as the encapsulation method for the ATM 1483 subinterface.

   host1(config-subif)#**encapsulation bridge1483**

3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the bridged Ethernet interface.

   host1(config-subif)#**encapsulation vlan**

4. Create a VLAN subinterface to carry the higher-level protocols by adding a subinterface number to the interface identification command.

   host1(config-subif)#**interface atm 3/1.1.1**

5. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-subif)#**svlan id 4 255**

6. Assign an S-VLAN Ethertype.

   host1(config-subif)#**svlan ethertype 9200**

Proceed to *Configuring Higher-Level Protocols over S-VLANs* on page 387 for information about configuring higher-level protocols over the S-VLAN subinterfaces.

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN subinterface.
- Choose one of the following Ethertype values:
  - 8100—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - 9100—Specifies Ethertype value 0x9100, which is the default
  - 9200—Specifies Ethertype value 0x9200

- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-if)#**svlan ethertype 8100**

- Use the **no** version to restore the default value, 9100.

svlan id

- Use to assign S-VLAN IDs and VLAN IDs to VLAN subinterfaces.
- Use S-VLAN ID and VLAN ID numbers that are in the range 0–4095 and that are unique within the Ethernet interface.
- Issue the **svlan id** command before any upper bindings are made, such as IP or PPPoE.
- Example

  host1(config-if)#**svlan id 4 255**

- There is no **no** version.

Configuring Higher-Level Protocols over S-VLANs

The procedures for configuring IP, PPPoE, and MPLS protocols over S-VLANs on bridged Ethernet interfaces are identical to the procedures for configuring these protocols over VLANs on bridged Ethernet interfaces.

This section provides an example for configuring PPPoE interfaces over S-VLAN subinterfaces configured on bridged Ethernet. For descriptions of the commands used in this procedure, see *Configuring Higher-Level Protocols over VLANs* on page 382.

To configure PPPoE over S-VLAN over a bridged Ethernet interface:

1. Follow the steps in *Configuring S-VLAN Subinterfaces over Bridged Ethernet* on page 386 to configure the S-VLAN subinterface.

2. Create a PPPoE major interface on the S-VLAN subinterface.

   host1(config-subif)#**pppoe**

3. Exit the subinterface context.

   host1(config-subif)#**exit**

4. Create a PPPoE subinterface by adding a subinterface number to the interface identification command.

   host1(config)#**pppoe subinterface atm 3/1.1.1.1**

5. Specify PPP as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation ppp**

6. Assign an IP address and mask to the interface.

   host1(config-subif)#**ip address 10.1.2.3 255.255.255.255**

Configuring the MTU Size for Bridged Ethernet

You can use the **bridge1483 mtu** command to configure a nondefault maximum transmission unit (MTU) size, in bytes, for a bridged Ethernet interface. The default MTU size for a bridged Ethernet interface is 1518 bytes.

Because you configure a bridged Ethernet interface over an ATM 1483 subinterface, the MTU size set with the **bridge1483 mtu** command is limited by the MTU set for the underlying ATM 1483 subinterface. As a result, the **bridge1483 mtu** command requires you to configure an MTU size for the bridged Ethernet interface that does not exceed the maximum allowable MTU size for the underlying ATM 1483 subinterface, 9180 bytes.

The configured MTU size for an interface is referred to as its *administrative MTU*, and the MTU size at which the interface actually operates is referred to as its *operational MTU*. For bridged Ethernet interfaces, the operational MTU is the lesser of the following two values:

- The administrative MTU of the bridged Ethernet interface
- The administrative MTU of the underlying ATM 1483 subinterface

You can also use the **bridge1483 mtu** command in a profile to configure a nondefault MTU size for a dynamic bridged Ethernet interface. For information, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

bridge1483 mtu

- Use to set the maximum allowable size, in bytes, of the MTU for bridged Ethernet interfaces.
- Specify an MTU size in the range 64–9180 bytes.
- Example

  host1(config-subif)#**bridge1483 mtu 1684**

- Use the **no** version to restore the default MTU size for bridged Ethernet interfaces, 1518 bytes.

Monitoring Bridged Ethernet

You can:

- Display information about bridged Ethernet interfaces by using the **show bridge1483 interface** command.
- Monitor MAC address validation by using the **show ip mac-validate interface** command.
- Display information about VLANs configured on bridged Ethernet interfaces by using the **show vlan subinterface** command.

Bridged Ethernet interfaces are not bound to a specific virtual router (VR).

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

show bridge1483 interface

- Use to display configuration and status information for all bridged Ethernet interfaces currently configured on the router.
- Use the **atm** keyword and an interface specifier to display information for a bridged Ethernet interface that is stacked over an ATM subinterface.
- Use the **summary** keyword to display only a count of all bridged Ethernet interfaces configured on the router.
- Field descriptions
  - Interface—Type and specifier of the lower-layer interface on which bridged Ethernet is configured
  - Status—Status of the bridged Ethernet interface: up, down, lowerLayerDown, notPresent
  - MAC Address—MAC address assigned to the bridged Ethernet interface, if configured
  - Type—Type of interface: static or dynamic
  - Oper/Admin MTU—Operational MTU, which is the MTU at which the interface actually operates, and administrative MTU, which is the MTU configured for the interface; the administrative MTU displays 1518 (the default value) if not configured
  - In—Analysis of inbound traffic on this interface
    - Bytes—Number of bytes received in error-free packets
    - Packets—Number of packets received
    - Multicast—Number of multicast packets received
    - Broadcast—Number of broadcast packets received

- Errors—Total number of errors in all received packets; some packets might contain more than one error
- Discards—Total number of discarded incoming packets

- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent
  - Packets—Number of packets sent
  - Multicast—Number of multicast packets sent
  - Broadcast—Number of broadcast packets sent
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- Total bridge1483 interfaces—Total number of bridged Ethernet interfaces configured on the router; this is the only information that appears when you specify the **summary** keyword

- Example 1—Displays full configuration and status information

```
host1#show bridge1483 interface
                                                                Oper/Admin
        Interface           Status    MAC Address    Type       MTU
------------------------ ---------- -------------- ------- ----------
ATM 5/1.1                Up         ----.----.---- Static  1500/1684
ATM 5/1.2                Up         ----.----.---- Static  8192/9188
2 bridge1483 interfaces found
```

- Example 2—Displays full status and configuration information for the specified bridge1483 interface

```
host1#show bridge1483 interface atm 12/0.1
                                                                Oper/Admin
        Interface           Status    MAC Address    Type       MTU
------------------------ ---------- -------------- ------- ----------
ATM 12/0.1               Up         ----.----.---- Static  1522/1522

  In: Bytes 0, Packets 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
  Out: Bytes 0, Packets 0
   Multicast 0, Broadcast 0
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

- Example 3—Displays only brief summary information

```
host1#show bridge1483 interface summary
Total bridge1483 interfaces: 3
```

show ip mac-validate interface

- Use to display the status of the MAC address validation on the physical interface.
- Field descriptions
  - *interfaceSpecifier*—Interface type slot/port
  - Keyword assigned to interface—Strict or Loose
  - Address—IP address of the entry
  - Hardware Addr—Physical (MAC) address of the entry
- Example

```
host1:vr1#show ip mac-validate interface atm 8/0.1
ATM8/0.1:  Strict

        Address     Hardware Addr
        180.1.0.2   0000.1111.2222
```

show vlan subinterface

- Use to display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router.
- Use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface. For more information about specifying the ATM physical interface on which you want to configure the VLAN subinterface, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - Status—Status of the VLAN subinterface: up, down, lowerLayerDown, notPresent
  - MTU—Maximum allowable size (in bytes) of the maximum transmission unit for the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID value for the VLAN subinterface
  - Ethertype—S-VLAN Ethertype value, if configured
  - Total VLAN interfaces—Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; this is the only field that appears when you specify the **summary** keyword

- Example 1—Displays full status and configuration information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype
------------------------ ---------- ---- -------- -------- ---------
ATM 3/0.1.2              Up         1522 ----     11       ----
ATM 3/0.1.3              Up         1522 ----     12       ----
ATM 3/1.1.1              Up         1522 ----     13       ----
ATM 3/1.1.2              Up         1522 ----     14       ----
ATM 3/2.1.1              Down       1526 4        255      0x9200
FastEthernet 4/5.1       Up         1522 ----     1        ----
6 vlan subinterfaces found
```

- Example 2—Displays full status and configuration information for the specified VLAN subinterface

```
host1#show vlan subinterface atm 3/0.1.2
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype
------------------------ ---------- ---- -------- -------- ---------
ATM 3/0.1.2              Up         1522 ----     11       ----
```

- Example 3—Displays only brief summary information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface summary
Total VLAN interfaces: 6 subinterfaces, 3 major interfaces
```

Chapter 13

Configuring Transparent Bridging

This chapter provides an introduction to transparent bridging and describes how to configure transparent bridging on E-series routers.

This chapter contains the following sections:

Overview

This section introduces important concepts that you need to understand before configuring transparent bridging. These concepts include:

- How Transparent Bridging Works
- Bridge Groups and Bridge Group Interfaces
- Bridge Interface Types and Supported Configurations
- Subscriber Policies
- Concurrent Routing and Bridging
- Transparent Bridging and VPLS
- Unsupported Features

How Transparent Bridging Works

A *transparent bridge* is a data-link layer (layer 2) relay device that connects two or more networks or network systems. When a transparent bridge powers up, it automatically begins learning the network topology by examining the media access control (MAC) source address of every incoming packet. The bridge then creates an entry in the forwarding table consisting of the address and associated interface where the packet was received.

More specifically, a transparent bridge performs all of the following actions to learn the network topology:

- Learning—The bridge examines the MAC address of every incoming packet, records the MAC address and associated interface in the forwarding table, and manages the database of MAC addresses and their associated interfaces.

- Flooding—When a packet's destination address does not match any entries in the forwarding table, the bridge transmits (floods) the packet on all bridge interfaces to all network segments except the interface on which the packet was received.

- Forwarding—Once the bridge has learned a packet's destination address (that is, has a matching entry in its forwarding table), the bridge uses the associated port and interface information to send the packet toward its destination.

- Filtering—If the bridge detects that a packet's source and destination addresses are on the same network segment, it ignores (filters) that packet. *Filtering* is the process by which the bridge can screen network traffic for certain characteristics and determine whether to forward or discard (drop) that traffic based on user-defined criteria. On E-series routers, filtering criteria can include the MAC source address, MAC destination address, and protocol type.

- Aging—When a bridge adds a dynamic (learned) MAC address entry to the forwarding table, it assigns an age to the entry. The bridge updates this age each time it receives a packet. To manage MAC entries more efficiently, you can configure an entry's aging time, which is the maximum time that an entry can remain in the forwarding table before it "ages out."

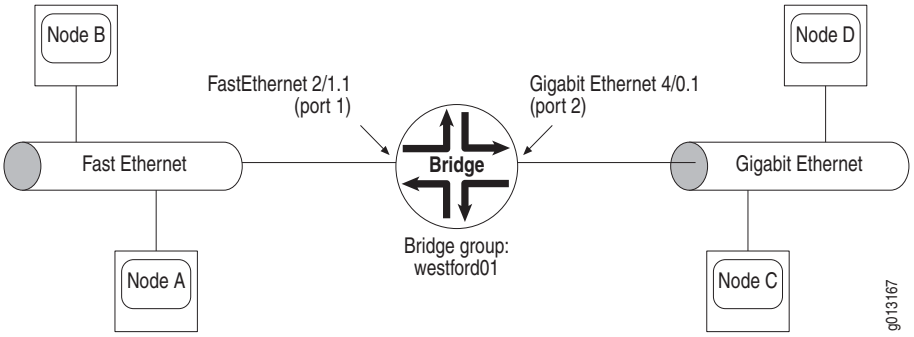
Bridge Groups and Bridge Group Interfaces

You configure transparent bridging by creating one or more bridge groups on the router. A *bridge group* is a collection of network interfaces (ports) that forms a broadcast domain. Each bridge group has its own set of forwarding tables and filters and, as such, functions as a logical transparent bridging device. For information about the maximum number of bridge groups that you can configure per E-series router, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

After you create a bridge group, you associate one or more network interfaces with the bridge group. This association is called a *bridge group interface*, or simply *bridge interface*. For information about the maximum number of bridge interfaces that you can configure per line module and per E-series router, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

Figure 38 shows an example of a simple transparent bridging network configuration that illustrates the concepts discussed so far in this section.

Figure 38: Bridge Group with Fast Ethernet and Gigabit Ethernet Bridge Interfaces



In Figure 38, a bridge group named westford01 is configured on the E-series router, which allows the router to function as a transparent bridge between a Fast Ethernet LAN segment and a Gigabit Ethernet LAN segment. The bridge group includes two bridge interfaces. The bridge interface associated with port 1 is stacked on a VLAN subinterface over a Fast Ethernet interface. The bridge interface associated with port 2 is stacked on a VLAN subinterface over a Gigabit Ethernet interface.

Table 21 presents a simple representation of the forwarding table for bridge group westford01.

Table 21: Sample Bridge Group Forwarding Table

Port	Source Address	Interface
1	Node A	Fast Ethernet 2/1.1
1	Node B	Fast Ethernet 2/1.1
2	Node C	Gigabit Ethernet 4/0.1
2	Node D	Gigabit Ethernet 4/0.1

Bridge Interface Types and Supported Configurations

A bridge interface can be configured as one of the following types:

- Subscriber (client)—A subscriber (client) bridge interface is *downstream* from the traffic flow; that is, the traffic flow direction is from the server (trunk) to the client (subscriber). This is the default bridge group interface type.

- Trunk (server)—A trunk (server) bridge interface is *upstream* from the traffic flow; that is, the traffic flow direction is from the client (subscriber) to the server (trunk). To configure a trunk bridge group interface, you must specify the **subscriber-trunk** keyword as part of the **bridge-group** command.

You can configure bridge interfaces to add transparent bridging capabilities to your existing network configurations. Currently, bridge interfaces can be stacked on:

- Bridged Ethernet over ATM 1483 subinterfaces
- Fast Ethernet interfaces
- Gigabit Ethernet interfaces
- 10-Gigabit Ethernet interfaces
- VLAN subinterfaces over Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, or bridged Ethernet interfaces

For sample configurations that include bridge interfaces, see *Configuration Examples* on page 413. For information about configuring Ethernet, ATM, and bridged Ethernet interfaces, see:

- *Chapter 1, Configuring ATM*
- *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces*
- *Chapter 12, Configuring Bridged Ethernet*
- *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*

Subscriber Policies

To enable intelligent flooding of packets within a bridge group's broadcast domain, each bridge group interface you create is associated with a default subscriber policy. A *subscriber policy* is a set of forwarding and filtering rules that defines how the bridge group interface handles various packet or attribute types, as follows:

- For each packet type, the subscriber policy specifies whether you want the bridge group interface to permit (forward) or deny (filter or drop) packets of that type.
- For the relearn attribute, the subscriber policy specifies whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table. Permit indicates that relearning is allowed, and deny indicates that relearning is prohibited.

The router provides two default subscriber policies: default Subscriber for subscriber (client) bridge interfaces, and default Trunk for trunk (server) bridge interfaces.

Table 22 lists the default values for each packet or attribute type defined in the default Subscriber and default Trunk policies. The only difference between the two policies is how broadcast packets and packets with unknown unicast destination addresses (DAs) are handled.

Table 22: Default Subscriber Policies for Bridge Group Interfaces

Packet/Attribute Type	Default Subscriber Policy	Default Trunk Policy
ARP	Permit	Permit
Broadcast	Deny	Permit
IP	Permit	Permit
MPLS	Permit	Permit
Multicast	Permit	Permit
PPPoE	Permit	Permit
Relearn	Permit	Permit
Unicast (user-to-user)	Permit	Permit
Unknown unicast DA	Deny	Permit
Unknown protocol	Permit	Permit

You cannot change the default subscriber policy values listed in Table 22 for a trunk bridge interface. You can, however, configure a nondefault subscriber policy for a subscriber bridge interface to change the default permit or deny value for one or more packet or attribute types. For details, see *Configuring Subscriber Policies* on page 406.

Concurrent Routing and Bridging

After you create the necessary bridge groups and bridge interfaces for your network configuration, you can use the **bridge crb** command to enable concurrent routing and bridging (CRB) for all bridge groups configured on your router. When CRB is enabled, the router can route a protocol among a group of interfaces in one bridge group and concurrently bridge the same protocol among a separate group of interfaces in a different bridge group on the router.

The router does not switch the protocol between the two bridge groups. Instead, it confines routed traffic to the routed interfaces and bridged traffic to the bridged interfaces. As a result, a protocol can be either routed or bridged on a particular interface, but cannot be both routed and bridged on the same interface.

By default, CRB is disabled for all bridge groups on the router. When you use the **bridge crb** command to enable CRB, it takes effect for all bridge groups currently configured on your router; you cannot enable CRB for some bridge groups on the router but not for others.

When you first enable CRB, the router issues an implicit **bridge route** command for any IP, MPLS, or PPPoE interfaces that are currently configured in the interface stack for the bridge group. This command directs the bridge group to route traffic for these protocols. After CRB has been enabled, you must issue an explicit **bridge route** command to route any new IP, MPLS, or PPPoE interface that is the first occurrence of this protocol in the bridge group. (See *Configuring Explicit Routing* on page 411 for details about using the **bridge route** command.)

As a result, it is important that you issue the **bridge crb** command after you configure all bridge group interfaces. In this way, the router can detect all IP, MPLS, or PPPoE interfaces in your configuration and direct the bridge group to route traffic from these protocols.

Transparent Bridging and VPLS

Except for the **bridge crb** and **bridge route** commands, you can use the existing transparent bridging commands to configure one or more instances of the Virtual Private LAN Service (VPLS), referred to as *VPLS instances*, on the router. VPLS employs a layer 2 virtual private network (VPN) to connect multiple individual LANs across a service provider's MPLS core network. The geographically dispersed multiple LANs functions as a single virtual LAN.

A single VPLS instance is analogous to a bridge group, and performs similar functions. In effect, a VPLS instance is a new or existing bridge group that has additional VPLS attributes configured.

For details about configuring and using VPLS, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 8, Configuring VPLS*.

Unsupported Features

The current E-series implementation of transparent bridging does not support the spanning-tree algorithm as defined in IEEE 802.1D.

NOTE: Because the spanning-tree algorithm is not currently supported, make sure that your topology avoids the creation of network loops.

Platform Considerations

You can configure transparent bridging on the following E-series routers:

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

Module Requirements

For information about the modules that support transparent bridging on ERX-14xx models, ERX-7xx 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 transparent bridging.

For information about the modules that support transparent bridging 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 transparent bridging.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the physical interface on which to configure transparent bridging. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about transparent bridging, consult the following resources:

- IEEE 802.1D—Media access control (MAC) bridges
- Draft Standard P802.1Q/D9 IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks
- RFC 1493—Definitions of Managed Objects for Bridges (July 1993)

Before You Configure Transparent Bridging

Before you configure transparent bridging on an E-series router, verify that:

- You have correctly installed a line module that supports transparent bridging. For a list of the line modules that support transparent bridging, see *ERX Module Guide, Appendix A, Module Protocol Support* or *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support transparent bridging.
- Each configured line can transmit data to and receive data from your switch connections.

Table 23 lists the prerequisite tasks for configuring transparent bridging and the resources that you can consult to learn how to perform these tasks.

Table 23: Prerequisite Tasks for Configuring Transparent Bridging

To Learn About	See
Preconfiguration and hardware diagnostic procedures	*ERX Hardware Guide* *E120 and E320 Hardware Guide*
Configuring T3 ATM line modules	*JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*
Configuring OCx/STMx ATM line modules	*JUNOSe Physical Layer Configuration Guide, Chapter 3, Configuring Unchannelized OCx/STMx Interfaces*
Configuring Ethernet line modules	*JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*

Also have the following information available:

- A diagram of your network topology indicating the names of the bridge groups and bridge group interfaces that you need to create
- On ERX-7xx models, ERX-14xx models, and ERX-310 routers, the slot and port numbers of the line modules over which you want to configure transparent bridging
- On E120 and E320 routers, slot, adapter, and port numbers of the IOAs over which you want to configure transparent bridging
- Types and specifiers for the interfaces and subinterfaces over which you want to create bridge group interfaces

Configuration Tasks

To configure transparent bridging on an E-series router:

1. Create a bridge group.
2. (Optional) Set optional attributes for the bridge group.
3. Configure bridge group interfaces.
4. (Optional) Configure nondefault subscriber policies for bridge interfaces.
5. (Optional) Enable concurrent routing and bridging.
6. (Optional) If CRB is enabled, configure explicit routing for IP, MPLS, or PPPoE protocols.

The following sections describe how to perform each of these tasks. See *Configuration Examples* on page 413 for detailed sample configurations.

NOTE: For information about the maximum values that the router supports for transparent bridging, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

Creating Bridge Groups

To create a bridge group:

1. From Global Configuration mode, create a bridge group and give it an alphanumeric name.

   host1(config)#**bridge westford01**

NOTE: Do not assign the bridge group the same name as an existing VR configured on your router.

2. (Optional) Repeat Step 1 to create additional bridge groups, one at a time.

   host1(config)#**bridge westford02**
   host1(config)#**bridge westford03**

3. (Optional) Use the appropriate **show** command to verify the bridge group creation.

   host1#**show bridge groups**

bridge

- Use to create a bridge group for transparent bridging.
- You must specify an alphanumeric name for the bridge group; the name can be a maximum of 32 characters and can use any combination of alphanumeric characters.

- Example

  host1(config)#**bridge westford04**

- Use the **no** version to remove the bridge group from the router.

Configuring Optional Bridge Group Attributes

After you create a bridge group, you can configure the following optional attributes for the bridge group to manage the MAC address entries in the bridge group's forwarding table:

- Enable or disable the bridge group's ability to acquire dynamically learned MAC addresses; acquiring dynamic MAC addresses is enabled by default.

  host1(config)#**bridge westford01 acquire**

- Enable or disable the bridge group's ability to filter (forward or discard) frames with a particular MAC source or destination address.

  host1(config)#**bridge westford01 address 0090.1a40.4c7c forward atm 3/0.1**
  host1(config)#**bridge westford02 address 1011.22c2.333d discard**

- Set the aging time of a dynamic (learned) entry in the forwarding table.

  host1(config)#**bridge westford01 aging-time 200**

- Set the maximum number of dynamic MAC addresses that a bridge group can learn.

  host1(config)#**bridge westford02 learn 10000**

You can also optionally enable SNMP link status processing for the bridge group. For example:

host1(config)#**bridge westford03 snmp-trap link-status**

bridge acquire

- Use to enable or disable a specified bridge group's ability to acquire dynamically learned MAC addresses; acquiring dynamic MAC addresses is enabled by default.
- Enables the bridge group to forward any frames it receives for nodes (stations) whose address it has learned dynamically.
- Example

  host1(config)#**bridge westford01 acquire**

- Use the **no** version to prevent the bridge group from acquiring dynamically learned MAC addresses and to limit forwarding only to those nodes that have a statically configured address entry in the forwarding table.

bridge address

- Use to enable or disable a specified bridge group's ability to filter (forward or discard) frames based on their MAC address.
- Enables the bridge group to filter frames by their MAC address and add static (nonlearned) address entries to the forwarding table.
- Specify the following:
  - *bridgeGroupName*—Alphanumeric name of the bridge group specified in the **bridge** command
  - *macAddress*—Unique 48-bit (6-byte) physical address or hardware address of the LAN network interface card as a dotted triple of four-digit hexadecimal numbers
- Specify one of the following filter types:
  - **forward**—Forwards frames destined for the specified MAC address out the specified interface
  - **discard**—Discards (drops) frames sent from or destined for the specified MAC address without further processing
- If you use the **forward** keyword, you must additionally specify the following:
  - *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide* for information
- Example 1—Forwards frames destined for the node with MAC address 0090.1a40.4c7c out the specified Fast Ethernet interface

  host1(config)#**bridge westford02 address 0090.1a40.4c7c forward fastEthernet 3/0.1**

- Example 2—Drops frames sent from or destined for the node with MAC address 1011.22b2.333c

  host1(config)#**bridge westford03 address 1011.22b2.333c discard**

- Use the **no** version to remove the static MAC address entry from the forwarding table.

bridge aging-time

- Use to set the length of time, in seconds, that a dynamic (learned) MAC address entry can remain in a specified bridge group's forwarding table.
- When a dynamic entry reaches its configured aging time, it "ages out" of the forwarding table.
- The default aging time is 300 seconds.

- The aging-time range is 1–1000000 seconds.
- Example

  host1(config)#**bridge westford04 aging-time 1000**

- Use the **no** version to restore the default value, 300 seconds.

bridge learn

- Use to set the maximum number of dynamic (learned) MAC address entries that a specified bridge group can learn.
- For information about the maximum number of learned MAC address entries combined for all bridge groups on an E-series router, see *JUNOSe Release Notes, Appendix A, System Maximums*.
- The default value is 0 (zero) learned addresses. This default implies that there is no maximum number of learned entries for an individual bridge group; that is, an individual bridge group can learn an unlimited number of MAC addresses, up to the maximum number that the router supports.
- Example

  host1(config)#**bridge westford05 learn 2000**

- Use the **no** version to restore the default value, 0 (zero) learned addresses.

bridge snmp-trap link-status

- Use to enable SNMP link status processing for a specified bridge group and to enable SNMP traps for all bridge interfaces configured in the bridge group.
- Example

  host1(config)#**bridge westford06 snmp-trap link-status**

- Use the **no** version to disable SNMP link status processing for the bridge group.

Configuring Bridge Group Interfaces

To configure a bridge group interface:

1. From Global Configuration mode, select the ATM, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface or subinterface that you want to assign to the bridge group.
2. Assign the interface or subinterface to an existing bridge group to create the bridge interface.
3. (Optional) Configure the bridge group interface as a trunk (server) interface.
4. (Optional) Enable SNMP link status processing for the bridge group interface.
5. (Optional) Set the maximum number of dynamic MAC addresses that the bridge group interface can learn.

For detailed sample configurations that include bridge interfaces, see *Configuration Examples* on page 413.

bridge-group

- Use to assign a bridge interface to an existing bridge group.
- To create a subscriber (client) bridge group interface, which is the default, you must supply the alphanumeric name of the bridge group (specified in the **bridge** command) to which you want to assign the interface.
- Optionally, you can also choose one of the following keywords:
  - **subscriber-trunk**—Creates a trunk (server) bridge group interface
  - **snmp-trap link-status**—Enables SNMP link status processing for the specified interface in the specified bridge group; SNMP link status processing is disabled by default
  - **learn** *addressCount*—Sets the maximum number of MAC addresses that the bridge group interface can learn, where *addressCount* is an integer in the range 0–64000. A value of 0 indicates that an individual bridge group interface can learn an unlimited number of MAC addresses, up to the maximum number that the router supports.
- Example 1—Creates a subscriber (client) bridge group interface for a bridge group named westford02 with SNMP link status processing enabled

  host1(config-subif)#**bridge-group westford02 snmp-trap link-status**

- Example 2—Sets the maximum number of learned MAC addresses on the westford02 bridge interface to 1000

  host1(config-subif)#**bridge-group westford02 learn 1000**

- Example 3—Creates a trunk (server) interface for a bridge group named westford03

  host1(config-subif)#**bridge-group westford03 subscriber-trunk**

- Use the **no** version to remove the interface from the bridge group and to restore the default value for the keyword you specified.

interface atm

- Use to select an ATM interface or subinterface type.
- Example

  host1(config)#**interface atm 3/2.1**

- Use the **no** version to remove the interface or subinterface.

interface fastEthernet

- Use to select a Fast Ethernet interface.
- Example

  host1(config)#**interface fastEthernet 1/0.2**

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

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

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

Configuring Subscriber Policies

To configure a nondefault client subscriber policy:

1. From Global Configuration mode, create the subscriber policy and assign it an alphanumeric name.

   host1(config)#**subscriber-policy client01**

   This command accesses Subscriber Policy Configuration mode.

2. From Subscriber Policy Configuration mode, define the rules for each packet or attribute type for which you want to change the default value. (All other packet or attribute types will continue to use the default values listed in Table 22 on page 397.)

   host1(config-policy)#**broadcast permit**
   host1(config-policy)#**multicast deny**
   host1(config-policy)#**relearn deny**

3. Exit Subscriber Policy Configuration mode.

   host1(config-policy)#**exit**

4. From Global Configuration mode, associate the new subscriber policy with the bridge group in which the subscriber (client) interface resides.

   host1(config)#**bridge westford02 subscriber-policy client01**

5. (Optional) Use the appropriate **show** commands to verify the creation of the subscriber policy and its association with the bridge group interface.

   host1#**show subscriber-policy client01**
   host1#**show bridge westford02**

arp

- Use to modify the subscriber policy for ARP to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) ARP packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- ARP packets are forwarded by default.
- Example

  host1(config-policy)#**arp deny**

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

bridge subscriber-policy

- Use to associate a subscriber (client) bridge interface with a nondefault subscriber policy.
- Specify the following:
  - *bridgeGroupName*—Alphanumeric name of the bridge group specified in the **bridge** command
  - *subscriberPolicyName*—Alphanumeric name of the subscriber policy specified in the **subscriber-policy** command
- Example

  host1(config)#**bridge westford02 subscriber-policy client01**

- Use the **no** version to remove the association with the subscriber policy.

NOTE: You cannot change the default subscriber policy values for a trunk (server) bridge interface. As a result, you cannot use the **bridge subscriber-policy** command to associate a nondefault subscriber policy with a trunk bridge interface.

broadcast

- Use to modify the subscriber policy for the broadcast protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) broadcast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Broadcast packets are filtered or dropped by default.
- Example

  host1(config-policy)#**broadcast permit**

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

ip

- Use to modify the subscriber policy for IP to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) IP packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- IP packets are forwarded by default.
- Example

  host1(config-policy)#**ip deny**

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

mpls

- Use to modify the subscriber policy for MPLS to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) MPLS packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- MPLS packets are forwarded by default.
- Example

  host1(config-policy)#**mpls deny**

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

multicast

- Use to modify the subscriber policy for the multicast protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) multicast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Multicast packets are forwarded by default.
- Example

  host1(config-policy)#**multicast deny**

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

pppoe

- Use to modify the subscriber policy for PPPoE to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) PPPoE packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- PPPoE packets are forwarded by default.
- Example

  host1(config-policy)#**pppoe deny**

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

relearn

- Use to modify the relearning policy for a subscriber (client) bridge interface.
- The **relearn** command defines whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table.
- Specify one of the following keywords:
  - **permit**—Enables relearning
  - **deny**—Prohibits relearning and forces the bridge interface to wait until an entry "ages out" of the forwarding table to relearn it on the new interface
- Relearning is enabled by default.
- Example

  host1(config-policy)#**relearn deny**

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

subscriber-policy

- Use to create a nondefault subscriber policy for a subscriber (client) bridge interface.
- A subscriber policy is a set of forwarding and filtering rules that defines how the bridge interface handles various packet types.
- You must specify an alphanumeric name for the subscriber policy; the name can be a maximum of 32 characters and can use any combination of alphanumeric characters.
- Example

  host1(config)#**subscriber-policy client01**

- Use the **no** version to remove the nondefault subscriber policy.

NOTE: You cannot change the default subscriber policy values for a trunk (server) bridge interface. As a result, you cannot use the **subscriber-policy** command to create a nondefault subscriber policy for a trunk interface.

unicast

- Use to modify the subscriber policy for the unicast (user-to-user) protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) unicast packets.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Unicast packets are forwarded by default.
- Example

  host1(config-policy)#**unicast deny**

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

unknown-destination

- Use to modify the subscriber policy for packets with unknown unicast DAs to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) packets with unknown unicast DAs.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Packets with unknown unicast DAs are filtered or dropped by default.
- Example

  host1(config-policy)#**unknown-destination permit**

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

unknown-protocol

- Use to modify the subscriber policy for packets containing an unknown protocol to define whether a subscriber (client) bridge interface permits (forwards) or denies (filters or drops) these packets.
- An unknown protocol is any protocol other than ARP, IP, MPLS, or PPPoE.
- Specify one of the following keywords:
  - **permit**—Forwards packets of this type
  - **deny**—Filters or drops packets of this type
- Packets containing an unknown protocol are forwarded by default.
- Example

  host1(config-policy)#**unknown-protocol deny**

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

Enabling Concurrent Routing and Bridging

To enable concurrent routing and bridging (CRB) for all bridge groups on the router:

1. From Global Configuration mode, issue the **bridge crb** command.

   host1(config)#**bridge crb**

2. (Optional) Use the appropriate **show** command to verify that CRB is enabled for the bridge groups on your router.

   host1#**show bridge groups details**

bridge crb

- Use to enable concurrent routing and bridging (CRB) for all bridge groups configured on an E-series router.
- CRB is disabled by default.
- When CRB is enabled, the router can route a protocol among a group of interfaces in one bridge group and concurrently bridge the same protocol among a separate group of interfaces in a different bridge group.
- The command takes effect for all bridge groups on an E-series router; you cannot enable CRB for some bridge groups on the router but not for others.
- Example

  host1(config)#**bridge crb**

- Use the **no** version to disable CRB on all bridge groups and restore the default bridging capability.

Configuring Explicit Routing

After you enable concurrent routing and bridging, you may need to issue the **bridge route** command to configure explicit routing for IP, MPLS, or PPPoE protocols if both of the following conditions are true:

- You configure new IP, MPLS, or PPPoE interfaces after you issue the **bridge crb** command to enable concurrent routing and bridging.
- The IP, MPLS, or PPPoE interface is the first occurrence of this protocol in the bridge group.

For example, assume that you want to route (rather than bridge) IP, MPLS, and PPPoE interfaces, but only IP and MPLS interfaces are configured when you issue the **bridge crb** command. The router detects the IP and MPLS interfaces and issues implicit **bridge route** commands to route these protocols.

If you subsequently add a new IP interface to a bridge group, you do not need to issue the **bridge route** command because the implicit **bridge route** command for IP is still in effect. However, if you subsequently add a new PPPoE interface to the bridge group, you must issue an explicit **bridge route** command for PPPoE to direct the bridge group to route PPPoE packets.

You can also use the **bridge route** command as a way to filter packets by routing. If you issue an explicit **bridge route** command for a protocol that is not currently configured in any of your bridge groups, the bridge group must route rather than bridge that protocol, but does not have the required interface stacking to do so. As a result, the bridge group discards (drops) those packets.

To configure explicit routing:

1. Ensure that you have enabled concurrent routing and bridging. (See *Enabling Concurrent Routing and Bridging* on page 411 for details.)

2. From Global Configuration mode, enable routing of IP, MPLS, or PPPoE packets in a specified bridge group.

   host1(config)#**bridge westford02 route ip**
   host1(config)#**bridge westford02 route mpls**
   host1(config)#**bridge westford03 route pppoe**

3. (Optional) Use the appropriate **show** command to verify that routing is enabled for the specified protocols in the bridge group.

   host1#**show bridge westford02**

bridge route

- Use to enable the routing of IP, MPLS, or PPPoE packets in a specified bridge group when concurrent routing and bridging (CRB) is enabled.
- If you issue this command for a protocol that is not configured in any bridge groups on your router, the bridge group discards (drops) those packets.
- You must specify the alphanumeric name of the bridge group specified in the **bridge** command.
- Choose one of the following keywords to indicate the protocol type that the bridge group routes: **ip**, **mpls**, or **pppoe**.
- Example

  host1(config)#**bridge westford02 route ip**

- Use the **no** version to disable routing of the specified protocol in the specified bridge group.

Configuration Examples

This section provides examples that show how to configure transparent bridging on the router. With each step, an illustration shows how the router is building the interface column.

Example 1: Bridging with Bridged Ethernet

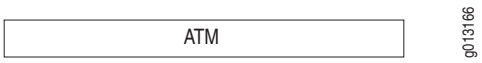
The following example illustrates how to configure transparent bridging with bridged Ethernet.

1. Create the bridge group.

   host1(config)#**bridge westford01**

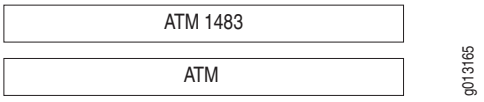
2. Create an ATM major interface.

   host1(config)#**interface atm 3/3**



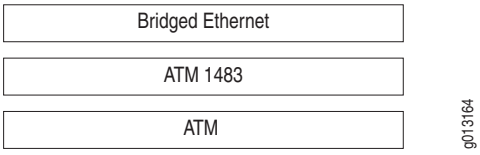
3. Create an ATM 1483 subinterface and associated PVC.

   host1(config-if)#**interface atm 3/3.1**
   host1(config-subif)#**atm pvc 1 0 10 aal5snap**



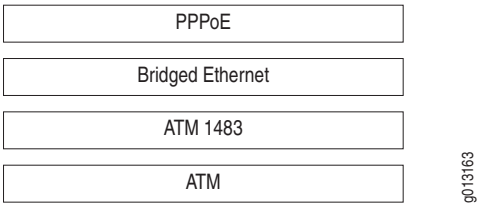
4. Specify bridged Ethernet as the encapsulation method on the subinterface. The **encapsulation** keyword implies that the bridged Ethernet interface is the only interface stacked directly above the ATM 1483 subinterface. As a result, the bridged Ethernet interface cannot have a peer interface stacked above the same lower-layer interface.

   host1(config-subif)#**encapsulation bridge1483**



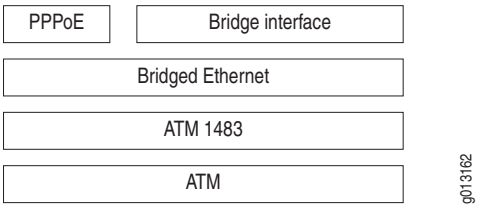
5. Create a PPPoE major interface over the bridged Ethernet interface. Because this command does not use the **encapsulation** keyword, the PPPoE interface can have one or more peer interfaces stacked above the same bridged Ethernet interface.

   host1(config-subif)#**pppoe**



6. Configure a subscriber (client) bridge group interface over the bridged Ethernet interface as a peer to the PPPoE interface. Assign the interface to the bridge group you created in Step 1.

   host1(config-subif)#**bridge-group westford01**



Example 2: Bridging with VLANs

The following example illustrates how to configure transparent bridging with VLANs over a Fast Ethernet interface.

NOTE: You can also configure transparent bridging with VLANs over a bridged Ethernet interface. For information, see *Configuring VLANs over Bridged Ethernet* in *Chapter 12, Configuring Bridged Ethernet*.

1. Create the bridge group.

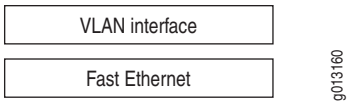
   host1(config)#**bridge westford02**

2. Create a Fast Ethernet interface.

   host1(config)#**interface fastEthernet 2/0**

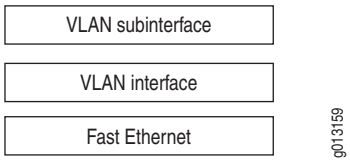
3. Create a VLAN major interface by specifying VLAN as the encapsulation method for the interface.

   host1(config-if)#**encapsulation vlan**



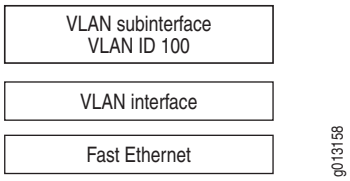
4. Create a VLAN subinterface by adding a subinterface number to the **interface fastEthernet** command.

   host1(config-if)#**interface fastEthernet 2/0.1**



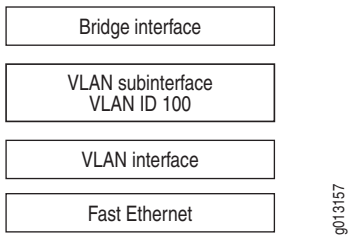
5. Assign a unique VLAN ID to the VLAN subinterface.

   host1(config-if)#**vlan id 100**



6. Configure a subscriber (client) bridge group interface over the VLAN subinterface. Assign the interface to the bridge group you created in Step 1.

   host1(config-subif)#**bridge-group westford02**



7. Exit Subinterface Configuration mode.

   host1(config-subif)#**exit**

8. (Optional) Configure additional VLAN subinterfaces and bridge group interfaces by repeating Steps 4 through 6, supplying unique values.

Monitoring Transparent Bridging

This section describes how to:

- Set a statistics baseline for bridge groups and bridge interfaces.
- Remove all dynamic MAC address entries or a specific dynamic MAC address entry from the forwarding table for bridge groups and bridge interfaces.
- Use the **show** commands to monitor bridge groups, bridge group interfaces, and subscriber policies

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

Setting Statistics Baselines

You can set a statistics baseline for a bridge group (by using the **baseline bridge** command) or for a bridge interface (by using the **baseline bridge interface** command).

baseline bridge

- Use to set a statistics baseline for a specified bridge group.
- 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.
- Example

  host1#**baseline bridge westford03**

- There is no **no** version.

baseline bridge interface

- Use to set a statistics baseline for a particular network interface belonging to a bridge group.
- 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.

- You must specify the following:
  - *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide* for information
- Example

  host1#**baseline bridge interface atm 3/3.1**

- There is no **no** version.

Removing Dynamic MAC Address Entries

You can remove all dynamic (learned) MAC address entries from the forwarding table for a bridge group (using the **clear bridge** command) or for a bridge interface (using the **clear bridge interface** command). You can also use the **clear bridge address** command to remove a specific dynamic MAC address entry from the forwarding table for a bridge group.

clear bridge

- Use to remove all dynamic MAC address entries from the forwarding table for the specified bridge group.
- Example

```
host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1               0
1234.abcd.5678   discard      ---                    ---

host1#clear bridge westford01

host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
```

- There is no **no** version.

clear bridge address

- Use to remove a specific dynamic MAC address entry from the forwarding table for the specified bridge group.
- Example

```
host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1               0
1234.abcd.5678   discard      ---                    ---

host1#clear bridge westford01 address 1234.abcd.5678

host1#show bridge westford01 table
  Bridge: westford01 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1               0
```

- There is no **no** version.

clear bridge interface

- Use to remove all dynamic MAC address entries for a network interface belonging to a bridge group from the forwarding table for that bridge group.
- You must specify the following:
  - *interfaceType*—One of the following bridge interface types listed in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide* for information
- Example

```
host1#show bridge westford02 table dynamic
  Bridge: westford02 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1               0
0090.1a01.0206   forward      ATM3/3.2              10
0090.1a01.0207   forward      ATM3/3.3               5

host1#clear bridge interface atm 3/3.2

host1#show bridge westford02 table dynamic
  Bridge: westford02 MAC Address Table
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
0090.1a01.0205   forward      ATM3/3.1               0
0090.1a01.0207   forward      ATM3/3.3               5
```

- There is no **no** version.

Monitoring Bridge Groups

You can use **show** commands to display information about the bridge groups configured on your router.

show bridge

- Use to display configuration and statistics information for the specified bridge group.
- To display information about the MAC address table and bridge interfaces, use the **all** keyword.
- Field descriptions
  - BridgeGroup—Name assigned to the bridge group
  - Bridge Mode—Bridging capability currently enabled, either concurrent routing and bridging (CRB) or default bridging
  - Aging Time—Length of time, in seconds, that a MAC address entry can remain in the forwarding table
  - Learning—Whether acquisition of dynamically learned MAC addresses is currently enabled or disabled
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group can learn
  - Link Status Snmp Traps—Whether SNMP link status processing is enabled or disabled for all bridge interfaces in the bridge group
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group
  - Protocol Actions—When CRB is enabled, displays the protocols (IP, MPLS, or PPPoE) for which explicit routing has been configured
  - Port Count—Number of ports (interfaces) currently configured for the bridge group; this value typically matches the Interface Count
  - Interface Count—Number of bridge group interfaces currently configured for the bridge group
  - Address Table—Displays the current static and dynamic entries in the MAC address table
    - Address—MAC address of the entry
    - Action—How the bridge group handles this entry: forward or discard
    - Interface—Interface type and specifier on which the entry will be forwarded; this value does not appear for entries that are discarded
    - Age—Length of time that a dynamic entry has been in the forwarding table; this value does not appear for static entries

- Interfaces—Displays statistics information for each bridge group interface; the entries for each interface are preceded by the interface type and specifier (for example, ATM3/3.1)
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
  - In Octets—Number of octets received on this interface
  - In Frames—Number of frames received on this interface
  - In Discards—Number of incoming packets discarded on this interface
  - In Errors—Number of incoming errors received on this interface
  - Out Octets—Number of octets transmitted on this interface
  - Out Frames—Number of frames transmitted on this interface
  - Out Discards—Number of outgoing packets discarded on this interface
  - Out Errors—Number of outgoing errors on this interface
  - queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
  - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped

- Example 1—Displays configuration settings for the specified bridge group

```
host1#show bridge westford01
  BridgeGroup: westford01
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             1
        Interface Count:        1
```

- Example 2—Displays information about configuration settings, MAC address table entries, and bridge group interfaces for the specified bridge group

```
host1#show bridge westford01 all
  BridgeGroup: westford01
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             1
        Interface Count:        1

  Address Table:
   Address         Action         Interface          Age
--------------   ----------   -------------------   -----
1011.22b2.333c   forward      ATM3/3.1               ---
1234.abcd.5678   discard      ---                    ---

  Interfaces:
    ATM3/3.1
      Port Number: 1
      Operational Status: LowerLayerDown
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: default Subscriber
      Statistics:
         In Octets:    0
         In Frames:    0
         In Discards:  0
         In Errors:    0
         Out Octets:   0
         Out Frames:   0
         Out Discards: 0
         Out Errors:   0
      queue 0: traffic class best-effort, bound to bridge     ATM3/3.1
        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
```

show bridge groups

- Use to display configuration information for all bridge groups currently configured on your router.
- To display the configuration settings for all bridge groups on your router, use the **details** keyword.
- Field descriptions
  - BridgeGroup—Name assigned to the bridge group
  - Bridge Mode—Bridging capability currently enabled, either concurrent routing and bridging (CRB) or default bridging
  - Aging Time—Length of time, in seconds, that a MAC address entry can remain in the forwarding table
  - Learning—Whether acquisition of dynamically learned MAC addresses is currently enabled or disabled
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group can learn
  - Link Status Snmp Traps—Whether SNMP link status processing is enabled or disabled for all bridge interfaces in the bridge group
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group
  - Protocol Actions—When CRB is enabled, displays the protocols (IP, MPLS, or PPPoE) for which explicit routing has been configured
  - Port Count—Number of ports (interfaces) currently configured for the bridge group; this value typically matches the Interface Count
  - Interface Count—Number of bridge group interfaces currently configured for the bridge group
- Example 1—Displays the names of the bridge groups configured on your router

```
host1#show bridge groups
  BridgeGroup: westford02
  BridgeGroup: westford01
```

- Example 2—Displays the configuration settings for each bridge group on your router

```
host1#show bridge groups details
  BridgeGroup: westford02
        Bridge Mode:            CRB
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      client01
        Protocol Actions:
           Route  IP
           Route  PPPoE
        Port Count:             0
        Interface Count:        0
```

```
BridgeGroup: westford01
      Bridge Mode:            CRB
      Aging Time:             300 secs
      Learning:               Enabled
      Max Learn:              Unlimited
      Link Status Snmp Traps: Disabled
      Subscriber Policy:      default Subscriber
      Protocol Actions:
      Port Count:             1
      Interface Count:        1
```

show bridge port

- Use to display configuration, statistics, and status information for all ports (interfaces) or for a specified port associated with a bridge group.
- To display only the port number, interface identifier, and status for each port, use the **brief** keyword.
- Field descriptions
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
  - Statistics—Displays statistics information for the specified port
    - In Octets—Number of octets received on this interface
    - In Frames—Number of frames received on this interface
    - In Discards—Number of incoming packets discarded on this interface
    - In Errors—Number of incoming errors received on this interface
    - Out Octets—Number of octets transmitted on this interface
    - Out Frames—Number of frames transmitted on this interface
    - Out Discards—Number of outgoing packets discarded on this interface
    - Out Errors—Number of outgoing errors on this interface
  - queue—Hardware packet queue associated with the specified traffic class and interface
    - Queue length—Length of the queue, in bytes
    - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
    - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped

- Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
- Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped

- Using the **brief** keyword displays the following fields:
  - Port—Bridge group port number on which this interface resides
  - Interface—Interface type and specifier associated with the port (for example, ATM3/3.1)
  - Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

- Example 1—Displays configuration, statistics, and status information for all ports currently associated with the bridge group

```
host1#show bridge westford01 port 1
    ATM3/3.1
      Port Number: 1
      Operational Status: LowerLayerDown
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: default Subscriber
      Statistics:
         In Octets:    0
         In Frames:    0
         In Discards:  0
         In Errors:    0
         Out Octets:   0
         Out Frames:   0
         Out Discards: 0
         Out Errors:   0
      queue 0: traffic class best-effort, bound to bridge     ATM3/3.1
        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 2—Uses the **brief** keyword to display summary information for each port

```
host1#show bridge westford01 port brief
 Port            Interface              Status
------   -------------------------   ------------
1        ATM3/3.1                    LowerLayerDown
```

show bridge table

- Use to display information about dynamic and static entries in the MAC address table for the specified bridge group.
- To display only static address entries, use the **static** keyword.
- To display only dynamic address entries, use the **dynamic** keyword.

- Field descriptions
  - Bridge—Name of the bridge group for which the MAC address table is displayed
  - Address—MAC address of the entry
  - Action—Specifies how the bridge group handles this entry: forward or discard
  - Interface—Interface type and specifier on which the entry will be forwarded; this value does not appear for entries that are discarded
  - Age—Length of time that a dynamic entry has been in the forwarding table; this value does not appear for static entries
- Example

```
host1#show bridge westford01 table static
  Bridge: westford01 MAC Address Table
   Address          Action          Interface           Age
--------------   ----------   -------------------   -----
1a11.22b2.333c   forward      ATM3/3.1               ---
1234.abcd.5678   discard      ---                    ---
```

Monitoring Bridge Interfaces

You can use the **show bridge interface** command to display information for a specified bridge interface or for all interfaces assigned to a bridge group.

show bridge interface

- Use to display configuration, statistics, and status information for a specified bridge interface or for all interfaces assigned to a bridge group.
- Field descriptions
  - BridgeGroup—Name of the bridge group to which the interface belongs
  - Port Number—Bridge group port number on which this interface resides
  - Operational Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
  - Admin Status—State of the physical interface: Up, Down
  - Snmp Link Status Trap—Whether SNMP link status processing is enabled or disabled for the specified bridge interface
  - Max Learn—Maximum number of dynamic MAC addresses that the bridge group interface can learn
  - Subscriber Policy—Name of the subscriber policy currently in effect for the bridge group interface
  - Statistics—Displays statistics information for the specified port
    - In Octets—Number of octets received on this interface
    - In Frames—Number of frames received on this interface
    - In Discards—Number of incoming packets discarded on this interface
    - In Errors—Number of incoming errors received on this interface
    - Out Octets—Number of octets transmitted on this interface

- Out Frames—Number of frames transmitted on this interface
- Out Discards—Number of outgoing packets discarded on this interface
- Out Errors—Number of outgoing errors on this interface

- queue—Hardware packet queue associated with the specified traffic class and interface
  - Queue length—Length of the queue, in bytes
  - Forwarded packets, Bytes—Number of packets and bytes forwarded on this queue
  - Dropped committed packets, Bytes—Number of committed packets and bytes that were dropped
  - Dropped conformed packets, Bytes—Number of conformed packets and bytes that were dropped
  - Dropped exceeded packets, Bytes—Number of exceeded packets and bytes that were dropped
- Using the **brief** keyword displays the following fields:
  - Interface—Interface type and specifier associated with the port (for example, FastEthernet9/1.1)
  - Port—Bridge group port number on which this interface resides
  - Status—Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

- Example 1—Displays information about a specified interface

```
host1#show bridge interface fastEthernet 9/1.1
    fastEthernet9/1.1
      BridgeGroup: 1
      Port Number: 1
      Operational Status: Up
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: atmfe1
      Statistics:
         In Octets:    0
         In Frames:    0
         In Discards:  0
         In Errors:    0
         Out Octets:   0
         Out Frames:   0
         Out Discards: 0
         Out Errors:   0
      queue 0: traffic class best-effort, bound to bridge
    FastEthernet9/1.1
        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 2—Uses the **brief** keyword to display a summary of all bridge interfaces configured on the router

```
host1#show bridge westford01 interface brief
Interface            Port       Status
-----------------  --------   ------------
FastEthernet9/1.1     1           Up
FastEthernet9/1.2     2           Up
FastEthernet9/3.1     3           Up
ATM11/0.5             4           Up
ATM11/3.2             5           Up
ATM11/0.7             6           Up
```

Monitoring Subscriber Policies

You can use the **show subscriber-policy** command to display the rules for all subscriber policies configured on your router or for a specified subscriber policy.

show subscriber-policy

- Use to display the set of forwarding and filtering rules for all default and nondefault subscriber policies configured on the router or for a specified subscriber policy.
- For all packet types except Relearn, the command displays **permit** to indicate that the bridge interface forwards the packets, or **deny** to indicate that the bridge interface filters the packets. (For information about the meaning of **permit** and **deny** for Relearn, see the field descriptions.)
- Field descriptions
  - Subscriber—Name of the subscriber policy
  - ARP—Specifies how the bridge interface handles ARP packets
  - Broadcast—Specifies how the bridge interface handles broadcast packets
  - Multicast—Specifies how the bridge interface handles multicast packets
  - Unknown Destination—Specifies how the bridge interface handles packets with unknown unicast DAs
  - Unicast—Specifies how the bridge interface handles unicast (user-to-user) packets
  - PPPoE—Specifies how the bridge interface handles PPPoE packets
  - Relearn—Specifies whether the bridge interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table; **permit** indicates that relearning is allowed, and **deny** indicates that relearning is prohibited
  - Mpls—Specifies how the bridge interface handles MPLS packets

- Example 1—Displays the rules for all default and nondefault subscriber policies currently configured on the router

```
host1#show subscriber-policy
  Subscriber: default Subscriber
  ARP                 : Permit
  Broadcast           : Deny
  Multicast           : Permit
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Permit
  Mpls                : Permit
  Subscriber: default Trunk
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Permit
  Unknown Destination : Permit
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Permit
  Mpls                : Permit

  Subscriber: client01
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Deny
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Deny
  Mpls                : Permit
```

- Example 2—Displays the rules for a specified subscriber policy

```
host1#show subscriber-policy client01
  Subscriber: client01
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Deny
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Deny
  Mpls                : Permit
```

Chapter 14
Configuring Cisco HDLC

Cisco High-Level Data Link Control (HDLC) is an encapsulation protocol that governs information transfer. This chapter describes how to configure Cisco HDLC on E-series routers.

This chapter contains the following sections:

Overview

Cisco HDLC is a bit-oriented synchronous data-link layer protocol developed by the International Organization for Standardization (ISO). It specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

By default, synchronous serial lines use the HDLC serial encapsulation method, which provides the synchronous framing-detection and error-detection functions of HDLC without windowing or retransmission.

Cisco HDLC monitors line status on a serial interface by exchanging keepalive request messages with peer network devices. It also enables routers to discover IP addresses of neighbors by exchanging Serial Line Address Resolution Protocol (SLARP) address-request and address-response messages with peer network devices.

The router responds to a SLARP address-request message from a remote peer with a SLARP address-response message, which indicates that it cannot participate in a SLARP session.

Cisco HDLC is compatible with Cisco Systems Cisco-HDLC protocol, the default protocol for all Cisco serial interfaces.

Framing

The router supports the following framing features:

- HDLC for data-link framing
- 18,000-byte information field size

Error Frames

All Cisco HDLC error frames are discarded.

SLARP Keepalive

One feature of Cisco HDLC is the exchange of keepalive messages. A keepalive message is a signal from one endpoint to the other that the first endpoint is still active. Keepalives are used to identify inactive or failed connections.

Platform Considerations

You can configure Cisco HDLC on the following E-series routers:

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

Module Requirements

For information about the modules that support Cisco HDLC on ERX-14xx models, ERX-7xx 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 Cisco HDLC.

For information about the modules that support Cisco HDLC 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 Cisco HDLC.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port* format to specify the physical interface on which you configure Cisco HDLC. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port* format. For example, the following command specifies a POS interface on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface pos 0/1
```

For E120 and E320 routers, use the *slot*/*adapter*/*port* format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies a POS interface on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface pos 5/0/0
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

Before You Configure Cisco HDLC

Before you configure a Cisco HDLC interface, you need to configure the physical interface over which Cisco HDLC traffic flows, described in the following chapters:

- *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*
- *JUNOSe Physical Layer Configuration Guide, Chapter 2, Configuring T3 and E3 Interfaces*

The procedures described in this chapter assume that a physical interface has been configured.

Configuration Tasks

To configure a Cisco HDLC interface:

1. Configure the physical interface on which you want to configure Cisco HDLC.

   host1(config)#**interface serial 3/1:2/1**

2. Select Cisco HDLC as the encapsulation method for the interface.

   host1(config-if)#**encapsulation hdlc**

3. Assign a local IP address to the interface.

   host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

4. (Optional) Use the appropriate **show hdlc interface** command to verify that the configuration changes are correct.

encapsulation hdlc

- Use to specify Cisco HDLC as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation hdlc**

- Use the **no** version to disable Cisco HDLC on the interface.

interface serial

- Use to configure a serial interface in the appropriate format by selecting a previously configured physical interface on which you want to configure Cisco HDLC. For example, to specify a channelized T3 interface, use the format *slot/port:channel/subchannel.subinterface*.
  - *slot*—Router chassis slot
  - *port*—Port on CT3, T3, or E3 I/O module
  - *channel*—T1 (DS1) channel
  - *subchannel—S*et of DS0 subchannels. For information about T1 subchannels, see *Fractional T1* in *JUNOSe Physical Layer Configuration Guide, Chapter 1, Configuring Channelized T3 Interfaces*.
  - *subinterface*—User-assigned number that identifies a subinterface
- Example

  host1(config)#**interface serial 3/1:2/1**

- Use the **no** version to remove the interface or subinterface.

ip address

- Use to assign an IP address and subnet mask to the interface.
- Example

  host1(config-subif)#**ip address 192.32.10.2 255.255.255.0**

- Use the **no** version to remove the IP address of the interface.

Optional Tasks

The following tasks are optional.

1. Configure the SLARP keepalive interval.

   host1(config-if)#**hdlc keepalive 10**

2. Enable loopback detection on an interface.

   host1(config-if)#**hdlc down-when-looped**

3. Disable an interface.

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

hdlc down-when-looped

- Use to enable loopback detection on a Cisco HDLC interface.
- By default, loopback detection is disabled.
- Example

  host1(config-if)#**hdlc down-when-looped**

- Use the **no** version to disable loopback detection on a Cisco HDLC interface.

hdlc keepalive

- Use to specify the keepalive timeout value.
- When the keepalive timer expires, the interface increments its own counter; then it compares the value of this counter with the last value received from a peer. If the difference between the values of the two counters is greater than three, the Cisco HDLC interface is declared down. After that, the interface sends a keepalive message containing the value of its counter and the last received value of the peer's counter.
- The router stores the values received in keepalive messages from a peer interface. If the interface is down, the router compares the received value of its own counter with the value from the peer. If the difference between the values of the two counters is less than four, the router declares the interface to be up. Both sides have to configure the same value for the keepalive interval.
- If the keepalive interval is 10 seconds, then a failed link is detected between 30 and 40 seconds after failure.
- The range is 0–6553 seconds. A value of 0 turns keepalive off.
- The default is 10 seconds.

- Example

  host1(config-if)#**hdlc keepalive 10**

- Use the **no** version to turn off the keepalive feature.

hdlc shutdown

- Use to terminate a Cisco HDLC session.
- This command administratively disables the interface.
- Example

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

- Use the **no** version to restart a disabled session. The default for each **hdlc shutdown** command is the **no** version.

Configuration Example

This example shows how to configure Cisco HDLC over an unchannelized DS3 interface on a cOCx/STMx line module. The example shows the complete configuration procedure, from configuring the SONET interface to assigning an IP address to the Cisco HDLC interface.

1. Create or select a virtual router, vr1.

   host1(config)#**virtual-router vr1**

2. Configure SONET controller, slot 13, port 0.

   host1:vr1(config)#**controller sonet 13/0**

3. Set the SONET clock source to internal.

   host1:vr1(config-controll)#**clock source internal module**

4. Create an OC1 path.

   host1:vr1(config-controll)#**path 1 oc1 1**

5. Create an unchannelized DS3 interface.

   host1:vr1(config-controll)#**path 1 ds3 1 unchannelized**

6. Set the DS3 interface clock source to internal.

   host1:vr1(config-controll)#**path 1 ds3 1 clock source internal module**

7. Exit Controller Configuration mode.

   host1:vr1(config-controll)#**exit**

8. Create or select a serial interface over the DS3 interface.

   host1:vr1(config)#**interface serial 13/0:1/1**

9. Set the encapsulation to Cisco HDLC.

   host1:vr1(config-if)#**encapsulation hdlc**

10. Enable loopback detection on the interface.

    host1:vr1(config-if)#**hdlc down-when-looped**

11. Assign an IP address to the interface.

    host1:vr1(config-if)#**ip address 160.1.0.1 255.255.255.0**

Monitoring Cisco HDLC

You can monitor Cisco HDLC interfaces using the **show hdlc interface** command.

You can set a statistics baseline for Cisco HDLC interfaces, subinterfaces, or circuits using the **baseline hdlc interface serial** command.

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

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

baseline hdlc interface

- Use to set a statistics baseline for Cisco HDLC 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.
- Example

  host1#**baseline hdlc interface serial 2/0:1/1**

- There is no **no** version.

show hdlc interface

- Use to display statistics for the specified HDLC interfaces.
- You can specify the following keywords:
  - **statistics**—Displays interface statistics
  - **delta**—Specifies that baselined statistics are to be shown
  - **status**—Displays the operational status of all configured interfaces
  - **closed**—Displays interfaces with administrative status Closed
  - **config**—Displays configuration information

- **down**—Displays interfaces with operational status Down
- **lower-layer-down**—Displays interfaces with operational status LowerLayerDown
- **not-present**—Displays interfaces with operational status NotPresent
- **open**—Displays interfaces with administrative status Open
- **up**—Displays interfaces with operational status Up
- **full**—Displays configuration information, status, and statistics
- **filter**—Specifies a CLI output filter

- Field descriptions
  - interface status—State of the interface:
    - Up—Traffic can flow on the interface
    - Down—Traffic cannot flow because of a problem in the interface at the current protocol layer
    - LowerLayerDown—Traffic cannot flow because of a problem in an interface at a lower protocol layer
    - NotPresent—Traffic cannot flow because hardware is unavailable
  - Interface administrative status—Configured state of the interface:
    - Open—**no hdlc shutdown** command is operative
    - Closed—**hdlc shutdown** command is operative
  - Interface maximum-transmission-unit—Configured MTU size
  - Interface keepalive time—Configured keepalive interval value
  - Interface loop detection—Status of loopback detection: enabled, disabled
  - Interface statistics:
    - packets in—Number of inbound packets received on the interface
    - packets out—Number of outbound packets transmitted on the interface
    - octets in—Number of inbound octets received on the interface
    - octets out—Number of outbound octets transmitted on the interface
    - errors in—Number of inbound errors received on the interface
    - errors out—Number of outbound errors transmitted on the interface
    - discards in—Number of inbound packets discarded on the interface
    - discards out—Number of outbound packets discarded on the interface

- Example 1

```
host1#show hdlc interface serial 5/1:5/1
Cisco-HDLC interface serial 5/1:5/1 is LowerLayerDown
```

- Example 2

```
host1#show hdlc interface full
Cisco-HDLC interface serial 4/0:2 is Up
Interface administrative status is open
Interface maximum-transmission-unit is 1596
Interface keepalive time is 10 seconds
Interface loop detection is disabled
Interface statistics              in                  out
   packets                        0                   0
   octets                         242                 242
   errors                         0                   0
   discards                       0                   0
Cisco-HDLC interface serial 5/0:1/1 is NotPresent
2 Cisco-HDLC interfaces found
```

Chapter 15

Configuring Dynamic Interfaces

This chapter explains upper-layer dynamic interfaces and describes the procedures for configuring them on E-series routers.

This chapter contains the following sections:

Overview

Before you begin configuring dynamic interfaces, review the concepts described in this section.

A *dynamic interface* is created automatically and transparently through some external event, typically through the receipt of data over a lower-layer link, such as an ATM virtual circuit (VC) or a virtual LAN (VLAN).

The layers of a dynamic interface are created based on the packets received on the link and can be configured through any one of the following:

- RADIUS authentication
- Profiles
- A combination of RADIUS authentication and profiles

You create and configure each layer of a *static interface* manually through an existing configuration mechanism such as the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

Unlike static interfaces, dynamic interfaces are not restored through nonvolatile storage (NVS) after a reboot.

Types of Dynamic Interfaces

There are two types of dynamic interfaces: upper-layer and bulk-configured. This chapter describes upper-layer dynamic interfaces, which enable you to dynamically create the following configurations:

- Dynamic IP, PPPoE, PPP, MLPPP, and bridged Ethernet interfaces over a static ATM 1483 interface
- Dynamic PPPoE subinterfaces over a static PPPoE major interface

Bulk-configured dynamic interfaces enable you to dynamically create ATM 1483 subinterfaces and VLAN subinterfaces by bulk-configuring a range of identifiers. For more information, see *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Autodetection

The router performs *autodetection,* also referred to as *autosensing,* to determine the layers of each dynamic interface. The autodetection process occurs when the router conditionally constructs interface layers based on the encapsulation type of the incoming packet.

Autodetection only uses system resources on demand based on what is detected in the incoming packet. Dynamic interfaces are created as a result of traffic on the interface. Dynamic interfaces can also be dynamically deleted without your intervention, thereby enabling any consumed system resources to be returned.

Unlike dynamic interfaces, static interfaces always allocate system resources upon creation, and always consume system resources, even when the interface is quiescent.

Upper-Layer Dynamic Interface Configurations

E-series routers support the following types of upper-layer dynamic interface configurations:

- Dynamic IP over static ATM 1483 (IPoA)
- Dynamic IP over dynamic PPP over static ATM 1483
- Dynamic IP over dynamic PPP over dynamic PPPoE over static ATM 1483
- Dynamic IP over dynamic bridged Ethernet over static ATM 1483
- Dynamic IP over dynamic MLPPP over static ATM 1483
- Dynamic IP over dynamic MLPPP over dynamic PPPoE over static ATM 1483
- Dynamic IP over dynamic PPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IP over dynamic MLPPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IP over dynamic MLPPP over dynamic PPPoE (with or without VLANs)

Internet Protocol version 4 (IPv4) is supported for all of these upper-layer dynamic interface configurations.

Currently, Internet Protocol version 6 (IPv6) is supported only when PPP or MLPPP is the layer immediately below the IPv6 layer in the interface column. Dynamic IPv6 is *not* supported directly over static ATM 1483, dynamic bridged Ethernet, or dynamic VLANs. Upper-layer dynamic interface columns that support IPv6 include the following:

- Dynamic IPv6 over dynamic PPP over static ATM 1483
- Dynamic IPv6 over dynamic MLPPP over static ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over static ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over static ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE subinterface over static PPPoE major interface (with or without VLANs)

For more information about IPv4, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*. For more information about IPv6, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 2, Configuring IPv6*.

Profiles

You can use profiles to configure dynamic interfaces. A *profile* is a set of characteristics that can be dynamically assigned to interfaces. By using a profile, you reduce the management of a large number of interfaces by applying a set of characteristics to multiple interfaces.

When you are configuring a large number of interfaces with the same attributes at the higher layers, you can use a profile to factor out all the common attributes of each layer into one place. This action affects one or more dynamic layers of the interface column. After you define the static lower layers, you assign a profile to the highest static layer of the interface column.

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

In contrast to static PPP interfaces (above which only dynamic IP interfaces can be created), static ATM 1483 subinterfaces support recognition and creation of the following upper dynamic interface types or *encapsulations*:

- Bridged Ethernet
- IP
- IPv6
- Multilink PPP
- PPP
- PPPoE

The **auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **auto-configure** command. For more information about using this command, see *auto-configure Command* on page 449.

RADIUS Authentication

RADIUS helps protect your network against unauthorized access. To accomplish this, RADIUS clients running on your router send authentication requests to a central RADIUS server. You can configure dynamic interfaces over interfaces through RADIUS authentication.

When a packet is received, the authenticating interface, either PPP or ATM 1483, establishes a session with RADIUS and passes the username and password to the RADIUS server. For dynamic IPoA or dynamic bridged Ethernet, the RADIUS username and password are obtained from the information specified by the **subscriber** command. The RADIUS server returns a grant or deny indication. If authentication is granted, the RADIUS attributes are returned, a user login is created, and the dynamic interfaces are configured from the RADIUS attributes.

ATM 1483 interfaces may receive configuration data from the RADIUS server in the form of *traffic-shaping* parameters.

Any changes made to a RADIUS configuration for a given dynamic interface do not take effect until an existing dynamic interface configured from this RADIUS entry is re-created, that is, deleted and then dynamically created.

ATM Oversubscription for Dynamic Interfaces

You can take advantage of oversubscription of static ATM 1483 subinterfaces and bulk-configured ATM VCs with the following dynamic interface configurations:

- The router supports oversubscription of static ATM 1483 subinterfaces when you configure the static ATM 1483 subinterface to support one of the following dynamic upper-layer encapsulation types: bridged Ethernet, IP, Multilink PPP, PPP, and PPPoE interfaces. For information about configuring dynamic upper-layer encapsulation types over a static ATM 1483 subinterface, see *About Configuring Dynamic Interfaces over Static ATM* on page 446.

- The router supports oversubscription of bulk-configured VC ranges when you create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface. For information about configuring dynamic ATM 1483 subinterfaces with bulk-configured VC ranges, see *Configuring ATM 1483 Dynamic Subinterfaces* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

How Oversubscription Works

Oversubscription is based on the capabilities of the ATM line module on which the dynamic interface is configured. For details about the capabilities of specific ATM line modules, see either *Module Capabilities* in *Chapter 1, Configuring ATM*, or the *Link Layer Maximums* tables in *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

Each ATM line module supports a maximum number of configured subinterfaces or VCs, and a smaller maximum number of subinterfaces or VCs that can be active at any one time. The maximum number of active subinterfaces or VCs determines the number of subscribers that can connect to the router through this line module at any one time.

As a result, you can oversubscribe static ATM 1483 subinterfaces or bulk-configured VC ranges by creating up to the maximum number of configured subinterfaces or VCs supported on the module, knowing that no more than the maximum number of active subinterfaces or VCs can be connected to the router at any one time.

Static ATM 1483 Subinterfaces

An active static ATM 1483 subinterface currently supports a dynamic upper-layer encapsulation type such as PPP or PPPoE. For ATM line modules that support ATM subinterface oversubscription, the maximum number of active subinterfaces supported per module is less than the maximum number of configured subinterfaces supported per module.

When the maximum number of active ATM 1483 subinterfaces has been reached, the router prevents all additional subscribers from connecting to the line module until at least one currently active subscriber logs out, which causes the router to tear down the dynamic interface column for that subscriber. When a dynamic interface column is torn down, the router enables the first currently inactive subscriber that receives traffic to connect to the router and become active as a replacement for the subscriber that logged out.

Example

Consider an ATM line module that supports a maximum of 16,000 configured subinterfaces and a maximum of 8000 active subinterfaces. If all 16,000 static ATM 1483 subinterfaces attempt to connect to the router, only the first 8000 subinterfaces to receive traffic are able to log in, generate dynamic interface columns, and become active. When a subscriber connected through one of these active subinterfaces logs out, the router enables the first of the remaining 8000 inactive subinterfaces that receives traffic to connect as a replacement for the subscriber that logged out.

Bulk-Configured VC Ranges

An active bulk-configured VC range is associated with a dynamic ATM 1483 subinterface that supports a dynamic upper-layer encapsulation type. For ATM line modules that support VC oversubscription, the maximum number of active bulk-configured VCs per line module is less than the maximum number of individual VCs created from the total number of bulk-configured VC ranges that the line module supports.

For details about how oversubscription works for bulk-configured VC ranges, see *ATM Oversubscription for Bulk-Configured VC Ranges* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges

ATM line modules are sometimes configured with a combination of static ATM 1483 subinterfaces and bulk-configured VC ranges. In these configurations, both the static ATM 1483 subinterfaces and bulk-configured VC ranges can support active subinterfaces. The combined total of active static ATM 1483 subinterfaces, and active dynamic ATM 1483 subinterfaces created from bulk-configured VC ranges, cannot exceed the maximum number of active subinterfaces supported by the line module.

For details about how oversubscription works for ATM modules configured with both static ATM 1483 subinterfaces and bulk-configured VC ranges, see *ATM Oversubscription for Bulk-Configured VC Ranges* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Platform Considerations

You can configure dynamic interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support dynamic interfaces on ERX-14xx models, ERX-7xx 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 dynamic interfaces.

For information about the modules that support dynamic interfaces 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 dynamic interfaces.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the physical interface that you want to configure to support dynamic interfaces. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

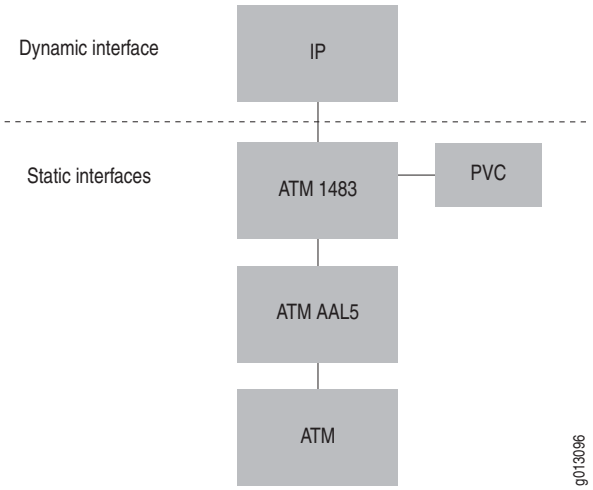
For more information about RADIUS, consult the following resources:

- RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)
- RFC 2866—RADIUS Accounting (June 2000)

About Configuring Dynamic Interfaces over Static ATM

To create dynamic interfaces over ATM, you create the static layers of the interface first, and then configure them to support a dynamic interface by means of autodetection. Figure 39 shows an example of the interface stack for a dynamic IP over ATM 1483 interface.

Figure 39: Configuring an ATM 1483 Interface to Support Dynamic Interfaces



On receipt of a packet, the router creates all dynamic layers above the ATM 1483 layer, starting with the lowest dynamic layer. For example, in the case of a dynamic PPPoE interface, the router creates the PPPoE interface first, then the PPP interface, and then the IP interface.

If any layer of the dynamic portion of the interface column fails to be created, then the interface creation fails and the connection is denied. All dynamic layers above the ATM 1483 subinterface are destroyed, starting with the highest dynamic layer.

When you configure a dynamic interface, you must assign (or create and assign) a profile to the interface. Profile creation and assignment topics are discussed in depth in *Configuring a Dynamic Interface from a Profile* on page 483.

About Configuring RADIUS for Dynamic Interfaces

Dynamic interfaces can be configured automatically through authentication and authorization by the RADIUS server.

On ATM interfaces, you initially create the static portion of the interface column by creating an ATM interface, ATM 1483 subinterface, and underlying ATM permanent virtual circuit (PVC).

subscriber Command

For dynamic interfaces that do not have a PPP layer, such as IPoA, you can use the **subscriber** command to configure an ATM 1483 subinterface to be authenticated automatically by the RADIUS server. The **subscriber** command uses a RADIUS username and optional password for identification and is available only for bridged Ethernet and IPoA configurations. This command is used for dynamic encapsulations that do not provide the authentication information remotely, as PPP does.

For dynamic interfaces with a PPP layer, the RADIUS username and password are obtained from the remote client, and authentication is performed with the RADIUS server. The attributes obtained from RADIUS can then be used to configure any higher-layer dynamic interfaces, such as IP, that are built over PPP.

For more information about using the **subscriber** command, see **subscriber** on page 476.

Authenticating Subscribers on Dynamic Bridged Ethernet over Static ATM Interfaces

You can use either of the following methods to configure and manage RADIUS authentication for IP subscribers on dynamic bridged Ethernet over static ATM interfaces:

- The **subscriber** command
- The subscriber management application

The **subscriber** command *does not support* running stateful SRP switchover (high availability) on the router. Therefore, the configuration method you choose depends on whether stateful SRP switchover is or is not running on your router.

Configuration Method Using subscriber Command

When you use the **subscriber** command to configure IP subscribers on dynamic bridged Ethernet over static ATM 1483 interface columns to support RADIUS authentication, the **subscriber** command provides the subscriber's authentication parameters. The static ATM 1483 subinterface acts as the authenticating layer that establishes a session with RADIUS and passes the subscriber's locally configured username and password information to the RADIUS server.

However, if your router is running stateful SRP switchover (high availability), the use of the **subscriber** command in this configuration might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. To bypass this limitation, you can use the subscriber management application to configure IP subscribers on dynamic bridged Ethernet interfaces.

Configuration Method Using Subscriber Management Application

You can use the JUNOSe subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet interface column. The subscriber management application uses an IP service profile to manage and authenticate IP subscribers with RADIUS. An IP service profile contains user and password information, and is used in a route map for subscriber management and to authenticate subscribers with RADIUS.

In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application acts as the authenticating layer to obtain information from RADIUS for configuration of dynamic IP subscribers. To assign the IP service profile to the interface profile from which the dynamic bridged Ethernet interface is created, you use the **bridge1483 service-profile** command in Profile Configuration mode.

If stateful SRP switchover is disabled or not running on your router, you can continue to use the **subscriber** command to configure IP subscribers on dynamic bridged Ethernet interfaces to support RADIUS authentication.

Alternatively, you can use the subscriber management application to create and configure dynamic IP interfaces regardless of whether stateful SRP switchover is running on the router. In addition, using subscriber management enables you to take advantage of several useful features such as the IP inactivity timer.

In the event that an interface profile for a dynamic bridged Ethernet interface includes the **subscriber** command to configure a local subscriber as well as the **bridge1483 service-profile** command to reference an IP service profile, the values specified with the **subscriber** command take precedence. The router ignores the values in the IP service profile in this case.

For details about using the subscriber management application to configure RADIUS authentication for IP subscribers on dynamic bridged Ethernet interfaces, see *Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces* on page 480.

For more information about using the subscriber management application, see *JUNOSe Broadband Access Configuration Guide, Chapter 23, Configuring Subscriber Management*.

Placing Dynamic IP Routes in the Routing Table

If you want to insert a dynamic IP route into the routing table of the relevant virtual router to point to the subscriber's subinterface, you can use the Framed-Route [22] RADIUS attribute to do so. Defined by *RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)*, the Framed-Route attribute can be returned in Access-Accept messages to specify the route as follows:

Framed-Route = *ipAddress/mask nextHop*

For dynamic IP interfaces, the next hop might not be known when you create the user record. In this case, use the value 0.0.0.0 for the next hop; the E-series router then assigns the subinterface associated with the user as the next hop in the routing table.

auto-configure Command

You use the **auto-configure** command to configure an ATM 1483 subinterface to support a dynamic interface. After the subinterface is configured, it performs autodetection to identify the encapsulation, resulting in the dynamic creation of the higher protocol layers. This command specifies one or more types of next upper dynamic encapsulations that the static interfaces can detect or accept.

NOTE: On static ATM 1483 interfaces, dynamic encapsulation types can be bridged Ethernet, IP, IPv6, PPP, or PPPoE.

Encapsulation Type Lockout

You can configure E-series routers to support dynamic encapsulation type lockout. With this feature, you can temporarily prevent an ATM 1483 subinterface from autodetecting, accepting, and creating dynamic interface columns for a configurable time period.

On ATM 1483 subinterfaces, encapsulation type lockout is the default behavior for IPoA, bridged Ethernet, PPP, and PPPoE encapsulation types.

Benefits

Using dynamic encapsulation type lockout provides the following benefits:

- Enables autodetection of other encapsulation types when a dynamic interface for a specified encapsulation type cannot be created.

  For example, when running a PPPoE client, DSL modems might transmit bridged Ethernet frames among the PPPoE frames. When bridged Ethernet and PPPoE encapsulation types are configured for autodetection with the **auto-configure** command, and a subscriber is configured for the bridged Ethernet encapsulation type, RADIUS sends a deny response after the router attempts to authenticate a received bridged Ethernet frame. Receiving an authentication denial from RADIUS causes the router to lock out bridged Ethernet. By locking out bridged Ethernet frames, the router can receive PPPoE frames unimpeded, facilitating rapid creation of dynamic PPPoE interfaces.

- Reduces loading on the RADIUS server.

  In some cases, IP and bridged Ethernet interfaces configured with a local subscriber do not have a corresponding subscriber entry in the RADIUS database. This can occur inadvertently due to misconfiguration of the E-series router or RADIUS server, or intentionally as a way to prevent creation of dynamic IPoA or bridged Ethernet interfaces.

  In previous releases, when the ATM 1483 interface received a deny response from RADIUS due to the missing subscriber entry, it performed continuous authentication retries every few seconds, which caused significant loading on the RADIUS server. Locking out autodetection of the IP or bridged Ethernet encapsulation type for a configurable time period prevents detection of dynamic IPoA or bridged Ethernet interfaces and reduces loading on the RADIUS server.

  For PPP and PPPoE encapsulation types, incorrect logins coupled with clients configured to perform frequent authentication retries results in significant loading on the RADIUS server. When an incorrect login occurs, the process of autodetecting, creating partial dynamic interface columns, and tearing down the columns due to authentication failures consumes router bandwidth. Enabling temporary lockout of PPP and PPPoE encapsulation types reduces loading on the RADIUS server caused by incorrect logins and auto-retry clients.

- Reduces loading on line modules.

  The repeated creation of multiple short-cycle dynamic interfaces causes excessive loading on line modules. A *short-cycle dynamic interface* is one that is detected, partially or completely created, and torn down within 60 seconds.

  Events that can cause short-cycle dynamic interfaces include:

  - Authentication denials from RADIUS due to the absence of a corresponding entry in the RADIUS database or due to improper login attempts
  - Misconfiguration within a dynamic interface profile or RADIUS record
  - Insufficient memory resources to create a dynamic interface column
  - Protocol failure or error that occurs within a dynamic interface column
  - Client logout shortly after a successful login; this action creates a complete dynamic interface column before the column is torn down

How Encapsulation Type Lockout Works

For a given encapsulation type, such as bridged Ethernet, lockout occurs when a dynamic interface of this type cannot be created. For example, an authentication denial from RADIUS causes a lockout. When lockout occurs, the router applies the lockout time range. If you do not configure a lockout-time range, the router uses the default time range.

Encapsulation type lockout is performed by default. You can configure the lockout time range by issuing the **auto-configure** command with the optional **lockout-time** keyword.

The following guidelines describe lockout behavior:

- Any encapsulation type that you do not configure for autodetection with the **auto-configure** command is automatically locked out.
- You can permanently lock out a specified encapsulation type from autodetection and prevent dynamic interface creation by issuing a **no auto-configure** command for the specified encapsulation type, if previously configured.
- When an encapsulation type is locked out, the router continues to autodetect the remaining encapsulation types and create the dynamic interfaces.

For the IP and bridged Ethernet encapsulation types, temporary lockout occurs automatically on receipt of an authentication deny response from RADIUS when you attempt to create and configure a dynamic IPoA or dynamic bridged Ethernet interface.

The lockout time range comprises two values: a minimum lockout time and a maximum lockout time. The initial lockout time begins with the minimum lockout time. From this point, the lockout time increases exponentially for every successive lockout event within the greater of 15 minutes or the maximum configured lockout time. The lockout time never exceeds the maximum value of the time range.

For example, using the default lockout time range of 1–300 seconds, the increasing lockout time sequence is: 1 second, 2 seconds, 4 seconds, 8 seconds, 16 seconds, 32 seconds, 64 seconds, 128 seconds, 256 seconds, and finally, 300 seconds (5 minutes).

Guidelines for Configuring Encapsulation Type Lockout

The following rules apply when you configure the lockout time for dynamic encapsulation type lockout:

- The lockout time value is defined as

  (minimum lockout time) * (2 ^ *n*-1)

  where *n* represents the number of consecutive lockout events.
- The router increments the value of *n* when the time between lockout events is either within 15 minutes or the maximum lockout time, whichever is greater.
- When the time between lockout events is greater than either 15 minutes or the maximum lockout time, the value of *n* reverts to 1. This condition is referred to as a *grace period*.
- The lockout time never exceeds the maximum configured lockout time. For example, for a configured lockout time in the range 20–120 seconds, the increasing lockout time sequence is 20 seconds, 40 seconds, 80 seconds, and finally, 120 seconds.

- A *short-cycle event* is a dynamic interface that is created and torn down within 60 seconds. The router tracks the time between short-cycle events to determine whether to increase the lockout time for a subsequent short-cycle event.

NOTE: When the calculated lockout time is equal to or exceeds the maximum lockout time, the router uses the maximum lockout time value until the time to the next event exceeds the greater of 15 minutes or the maximum lockout time value. At that point, the lockout time reverts to the minimum lockout time value.

- The minimum lockout time value cannot exceed the maximum lockout time value. When the minimum and maximum values are equal, the encapsulation type lockout time becomes fixed.

atm pvc Command

You use the **atm pvc** command to define the underlying circuit supporting an ATM 1483 subinterface. When you define a circuit with this command by using the **aal5autoconfig** option, it causes the ATM 1483 encapsulation (LLC/SNAP encapsulation or VC multiplexed) to be autodetected. Alternatively, if you use the **aal5snap** or **aal5mux ip** option, the ATM 1483 encapsulation becomes fixed, but higher layers can be dynamic.

For example, the following command configures a circuit for autodetection of the ATM 1483 encapsulation and all higher layers.

```
host1(config-subif)#atm pvc 100 0 100 aal5autoconfig 0 0 0
```

You can also include the **atm pvc** command in a base profile assigned to a dynamic ATM 1483 interface to apply encapsulation and traffic-shaping parameters to a bulk-configured range of PVCs. For information, see *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Configuring PPP and PPPoE Dynamic Interfaces over Static ATM

E-series routers support dynamic PPP and PPPoE interfaces. The configuration procedure is very similar for each.

When using the **auto-configure** command, select only **ppp** or **pppoe**. The router automatically builds the necessary interfaces for you. When you indicate **pppoe**, on receipt of a PPPoE packet, the dynamic interface built is IP over PPP over PPPoE over ATM. Likewise, when you indicate **ppp**, the dynamic interface built is IP over PPP over ATM.

Figure 40 shows dynamic PPP interface columns on ATM interfaces.

Figure 40: Dynamic PPP Interface Columns

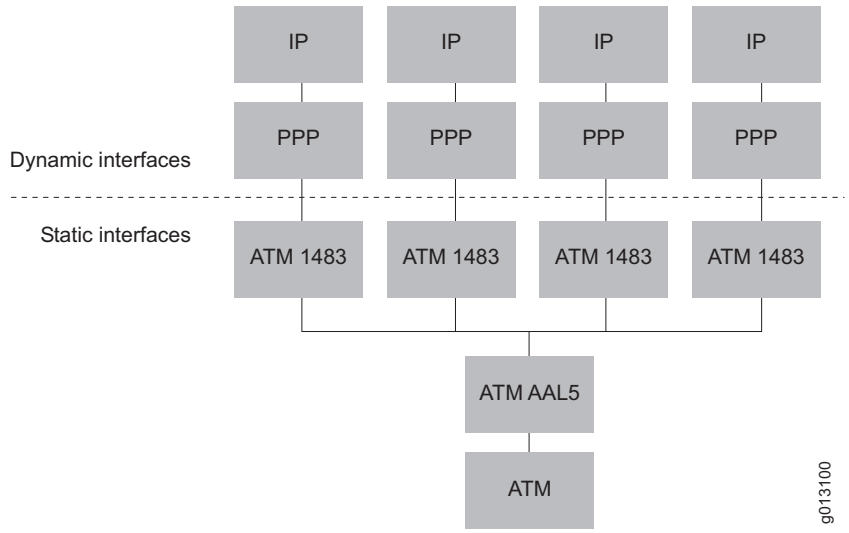
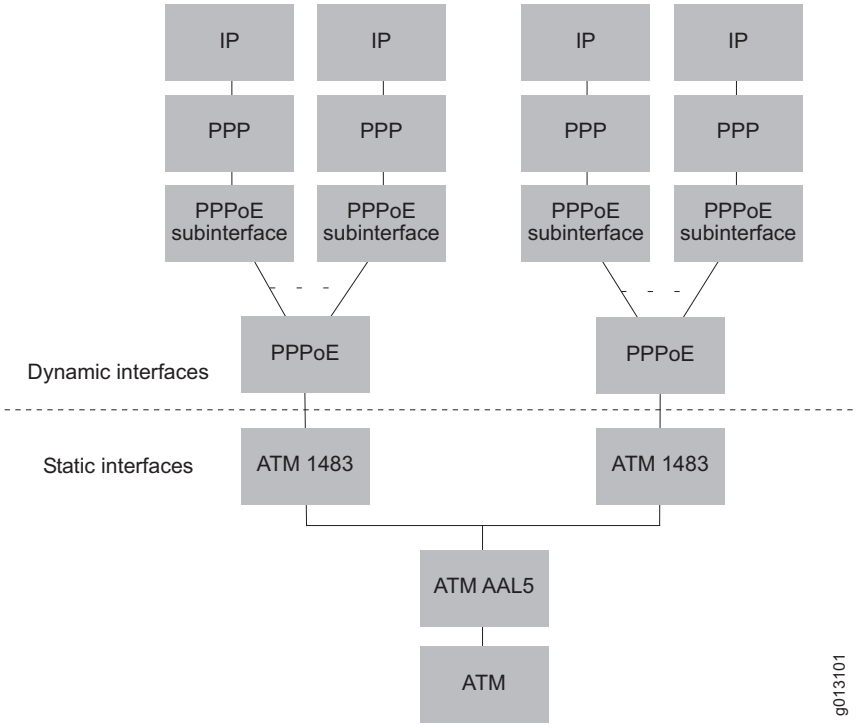


Figure 41 shows dynamic PPPoE interface columns and illustrates how PPPoE supports multiple IP sessions over each ATM 1483 circuit.

Figure 41: Dynamic PPPoE Interface Columns



You can specify either or both **ppp** and **pppoe** for the interface by specifying the **auto-configure** command for each type of interface. The first packet received defines the type of dynamic interface that is created.

Configuring a PPP or PPPoE Dynamic Interface

To configure an ATM 1483 subinterface to support a PPP or PPPoE dynamic interface:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 5/0.1**

3. Configure a PVC by specifying the virtual circuit descriptor (VCD), the virtual path identifier (VPI), the virtual channel identifier (VCI), and the encapsulation type.

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**
   host1(config-subif)#**atm pvc 10 100 22 aal5autoconfig**

4. Assign a profile to the PPP or PPPoE encapsulation types.

   host1(config-subif)#**profile ppp foo**
   host1(config-subif)#**profile pppoe foo**

5. Configure the subinterface to detect and accept dynamic PPP or PPPoE.

   host1(config-subif)#**auto-configure ppp**
   host1(config-subif)#**auto-configure pppoe**

   In addition to **ppp** and **pppoe**, you can also specify **ip** or **bridgedEthernet**.

6. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 5/0.1**

atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed).
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; the LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.

- Example

  host1(config-subif)#**atm pvc 6 0 11 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

auto-configure

- Use to configure a static ATM 1483 subinterface to support a dynamic interface. Specifies the types of dynamic encapsulation that the subinterface detects and accepts.
- This command creates the layers above ATM 1483 *dynamically*.
- You can enter the command repetitively to support multiple dynamic interface types.
- Select the dynamic next upper-interface type from these options: **bridgedEthernet**, **ip**, **ppp**, or **pppoe**.
- Encapsulation type lockout is performed on a per-encapsulation-type basis for each subinterface. An encapsulation type not configured for autodetection with the **auto-configure** command is automatically locked out. The lockout temporarily prevents the static ATM 1483 subinterface from detecting, accepting, and creating the encapsulation type until the lockout time expires.
- Use the **lockout-time** keyword to set the minimum lockout time and maximum lockout time, each of which can be in the range 1–86400 seconds (24 hours).The default range is 1–300 seconds (5 minutes).
- Use the **none** keyword to disable lockout for the specified encapsulation type.

NOTE: Disabling lockout can result in undesirable CPU loading; we recommend that you not disable lockout for general use. At a minimum, use the default lockout time.

- For information about the rules that apply when you configure the lockout time for dynamic encapsulation type lockout, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451.
- Example 1—Enables autodetection for the PPPoE encapsulation type using the default lockout time range, 1–300 seconds

  host1(config-subif)#**auto-configure pppoe**

- Example 2—Enables autodetection for the PPP encapsulation type using a nondefault lockout time range, 5–60 seconds

  host1(config-subif)#**auto-configure ppp lockout-time 5 60**

- Example 3—Disables encapsulation type lockout for the PPPoE encapsulation type

  host1(config-subif)#**auto-configure pppoe lockout-time none**

- Example 4—Either command reenables encapsulation type lockout for the PPPoE encapsulation type using the default lockout time range

  host1(config-subif)#**auto-configure pppoe**
  host1(config-subif)#**no auto-configure pppoe lockout-time**

- Example 5—Permanently locks out the PPP encapsulation type until the **auto-configure ppp** command is issued

  host1(config-subif)#**no auto-configure ppp**

- Use the **no** version to terminate detection of the specified encapsulation type or, if the **lockout-time** keyword is specified, to restore the lockout time range to its default value, 1–300 seconds.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 5/0/0.1**

- Use the **no** version to remove the interface or subinterface.

profile

- Use to assign a profile.
- You must specify the encapsulation type to which the profile applies: **bridgedEthernet**, **ip**, **ppp**, **pppoe**, or **any**.
- Specify a profile name with up to 80 alphanumeric characters.
- Example

  host1(config-subif)#**profile ppp foo**

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

Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations

In configurations of dynamic IP over dynamic PPP over a static ATM 1483 subinterface, as shown in Figure 40 on page 453, any of the following conditions might cause the static ATM 1483 subinterface to transition to a dormant state as the result of an ungraceful subscriber logout:

- Rebooting the router
- Rebooting a line module
- Transitioning the physical (for example, SONET) interface, ATM major interface, or ATM AAL5 interface from up to down to up again
- Transitioning the ATM 1483 subinterface or the ATM PVC from up to down to up again
- Any other lowerLayerDown operational status condition that affects the dynamic PPP interface; a lowerLayerDown status indicates that a lower-layer interface below the dynamic PPP interface is down

When the ATM 1483 subinterface transitions to a dormant state as a result of any of these conditions, the router tears down the dynamic PPP interface column. The dynamic PPP interface is unable to send an LCP terminate request to its peer because its own lower-layer interface is down. This action causes a loss of connectivity between the router and the PPPoA customer premises equipment (CPE). If the CPE supports the PPP keepalive feature, it can detect the loss of connectivity and restart Link Control Protocol (LCP) negotiations in order to initiate a new connection. However, if the CPE does not support PPP keepalive, it cannot detect that the connection is down, and continues to send PPP data packets to the router.

On receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet from the CPE when the ATM 1483 subinterface transitions to a dormant state, the router sends an LCP terminate request packet to the CPE. Receipt of the LCP terminate request packet causes the CPE to restart LCP negotiations in order to initiate a new connection. After the CPE restarts LCP negotiations, the router recreates the dynamic PPP and IP upper-layer interfaces above the static ATM 1483 subinterface. This behavior is always in effect on the router and does not require CLI or SNMP configuration.

Sending an LCP terminate request packet in response to receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet from a PPPoA CPE device offers the following benefits:

- For CPEs that support PPP keepalive, receipt of an LCP terminate request packet from the router restarts the LCP negotiations more quickly.
- For CPEs that do not support PPP keepalive, receipt of an LCP terminate request packet from the router enables the CPE to detect the connection termination and restart LCP negotiations in response.

The router also sends an LCP terminate request packet to a PPPoA CPE device in configurations of dynamic IP over dynamic PPP over a dynamic (bulk-configured) ATM 1483 subinterface. For more information, see *Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations* in *Chapter 16, Configuring Dynamic Interfaces Using Bulk Configuration*.

Configuring PPPoE Dynamic Interfaces over PPPoE Static Interfaces

E-series routers support dynamic PPPoE subinterfaces over static PPPoE major interfaces. The PPPoE major interfaces can be created over:

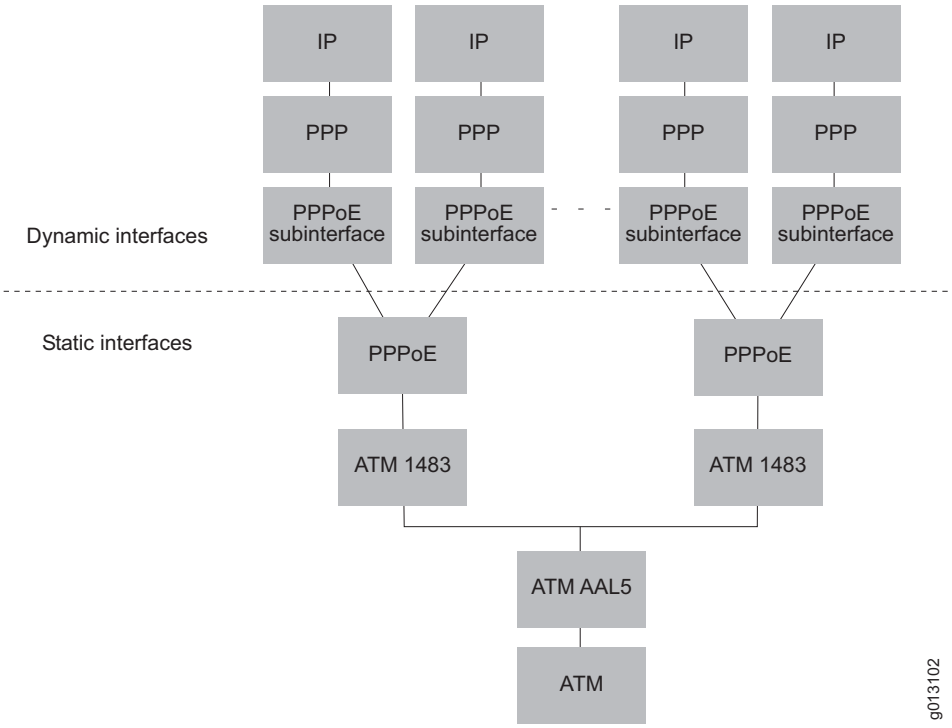
- ATM
- Ethernet
- Ethernet with VLANs
- Ethernet with S-VLANs

The following sections describe how to create each of these configurations on the router. In addition, *Configuring Encapsulation Type Lockout for PPPoE Clients* on page 467 describes how to configure dynamic encapsulation type lockout for PPPoE clients associated with dynamic PPPoE subinterface columns.

Configuring Dynamic PPPoE over Static PPPoE with ATM Interface Columns

Figure 42 shows dynamic PPPoE subinterface columns and illustrates an alternative method for PPPoE to support multiple IP sessions over each ATM 1483 circuit.

Figure 42: Dynamic PPPoE over Static PPPoE with ATM Interface Columns



To configure an ATM 1483 subinterface to support a dynamic PPPoE subinterface:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM 1483 subinterface.

   host1(config-if)#**interface atm 5/0.1**

3. Configure a PVC by specifying the virtual circuit descriptor (VCD), the virtual path identifier (VPI), the virtual channel identifier (VCI), and the encapsulation type.

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**
   host1(config-subif)#**atm pvc 10 100 22 aal5autoconfig**

4. Set the encapsulation type to PPPoE to create the PPPoE major interface.

   host1(config-subif)#**encapsulation pppoe**

5. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile1**

6. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

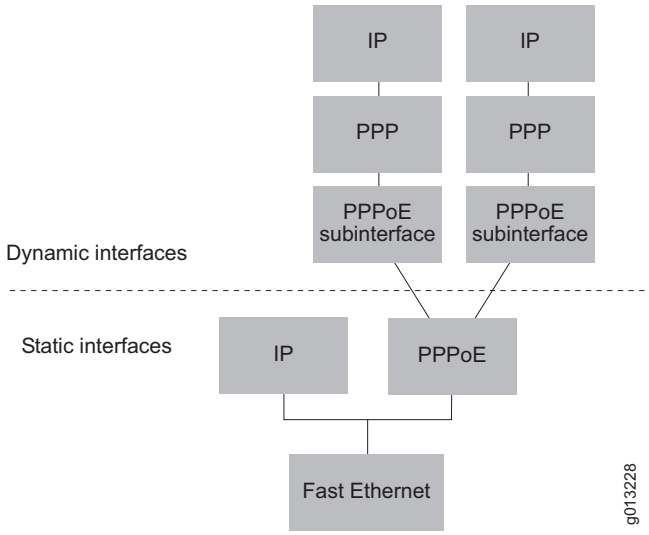
7. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 5/0.1**
   host1#**show pppoe interface atm 5/0.1**

Configuring Dynamic PPPoE over Static PPPoE with Ethernet Interface Columns

Figure 43 shows dynamic PPPoE subinterface columns configured over an Ethernet interface without VLANs.

Figure 43: Dynamic PPPoE over Static PPPoE with Non-VLAN Interface Columns



To configure an Ethernet interface without VLANs to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Assign an IP address and mask.

   host1(config-if)#**ip address 192.6.129.5 255.255.255.0**

3. Specify PPPoE as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation pppoe**

   This command creates the static PPPoE major interface.

4. Assign a profile to the PPPoE major interface.

   host1(config-subif)#**pppoe profile pppoeProfile3**

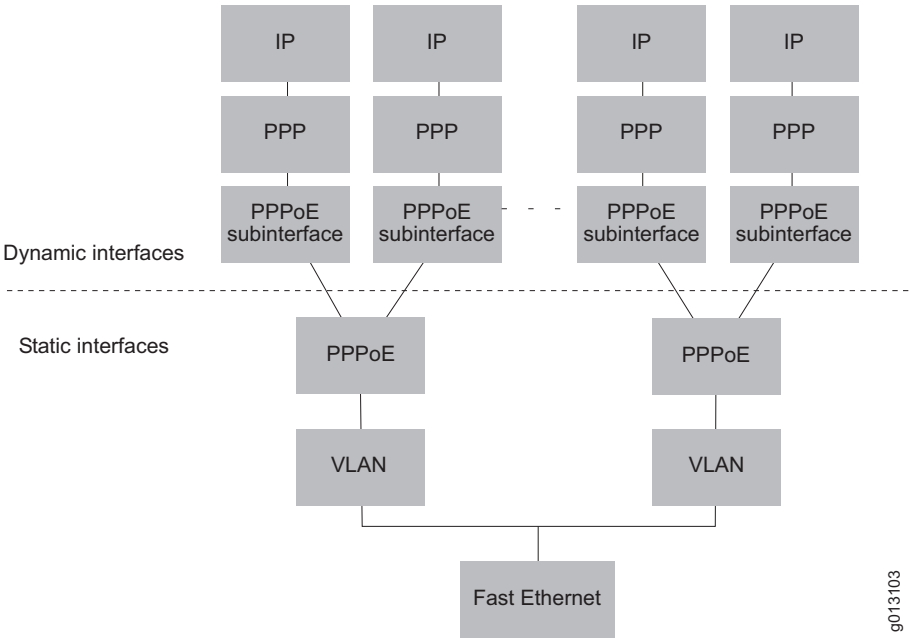
5. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

Configuring Dynamic PPPoE over Static PPPoE with Ethernet and VLAN Interface Columns

Figure 44 shows dynamic PPPoE subinterface columns and illustrates an alternative method for PPPoE to support multiple IP sessions over each VLAN.

Figure 44: Dynamic PPPoE over Static PPPoE with VLAN Interface Columns



To configure a VLAN subinterface to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   This command adds the VLAN major interface.

3. Create a VLAN subinterface by adding a subinterface number to the interface identifier.

   host1(config-if)#**interface fastEthernet 4/1.1**

4. Assign a VLAN ID for the subinterface.

   host1(config-if)#**vlan id 400**

5. Set the encapsulation type to PPPoE.

   host1(config-subif)#**encapsulation pppoe**

6. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile2**

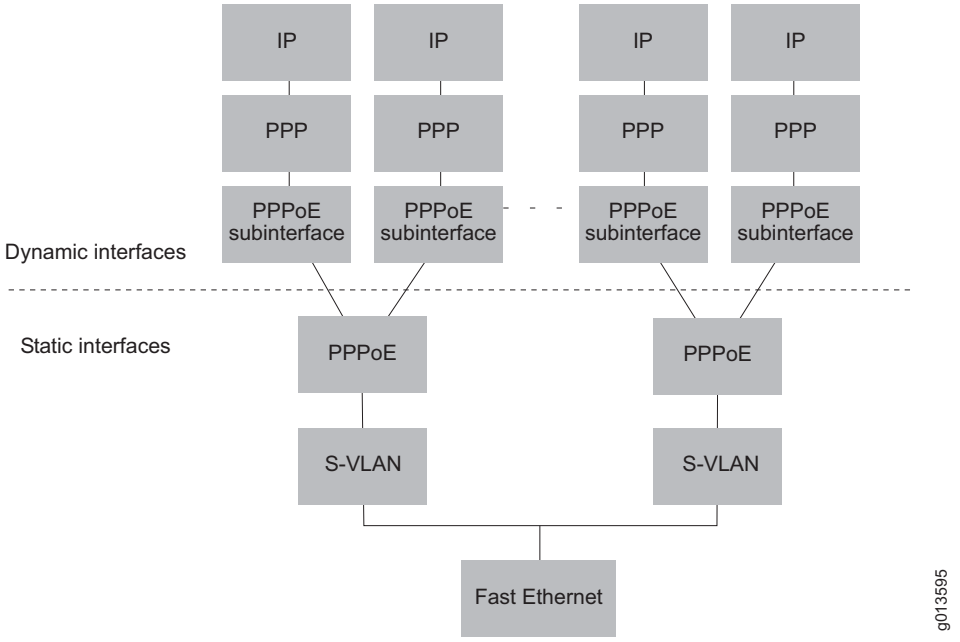
7. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

Configuring Dynamic PPPoE over Static PPPoE with Ethernet and S-VLAN Interface Columns

Figure 45 shows dynamic PPPoE subinterface columns over PPPoE major interfaces using S-VLANs over Ethernet.

Figure 45: Dynamic PPPoE over Static PPPoE with S-VLAN Interface Columns



To configure an S-VLAN subinterface to support a dynamic PPPoE subinterface:

1. Specify a Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interface.

   host1(config)#**interface fastEthernet 4/1**

2. Specify VLAN as the encapsulation method.

   host1(config-if)#**encapsulation vlan**

   This command creates the VLAN major interface.

3. Create a VLAN subinterface by adding a subinterface number to the interface identifier.

   host1(config-if)#**interface fastEthernet 3/1.1**

4. Assign an S-VLAN ID and a VLAN ID for the subinterface.

   host1(config-if)#**svlan id 3 300**

5. Assign an S-VLAN Ethertype.

   host1(config-if)#**svlan ethertype 9200**

6. Specify PPPoE as the encapsulation method on the interface.

   host1(config-subif)#**encapsulation pppoe**

   This command creates the PPPoE major interface.

7. Assign a profile.

   host1(config-subif)#**pppoe profile pppoeProfile3**

8. Configure the interface to detect and accept dynamic PPPoE subinterfaces.

   host1(config-subif)#**pppoe auto-configure**

S-VLAN Oversubscription

When you configure S-VLAN subinterfaces over Ethernet interfaces to support dynamic PPPoE subinterfaces, you can take advantage of S-VLAN oversubscription.

The maximum number of S-VLANs that you can create per I/O module or IOA with PPPoE major interfaces stacked over them is greater than the maximum number of dynamic PPPoE subinterfaces. The maximum number of PPP interfaces supported per line module is directly proportional to the maximum number of PPPoE subinterfaces.

As a result, you can oversubscribe S-VLANs by configuring up to the maximum number of S-VLANs supported on the I/O module or IOA, knowing that no more than the maximum number of supported PPP sessions can be connected to the router at any one time.

For information about the module combinations that support S-VLAN oversubscription, see *S-VLAN Oversubscription* in *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces*.

For specific information about the maximum number of S-VLANs supported per I/O module or IOA and the maximum number of PPP interfaces and PPPoE subinterfaces supported per line module, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: S-VLAN oversubscription is not currently supported for S-VLANs configured over bridged Ethernet interfaces.

NOTE: The E120 and E320 routers can support up to two IOAs per line module. This maximum number of S-VLANs per line module does not change whether one or two IOAs are installed. For more information about configuration options for the ES2-S1 GE-4 IOA, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.

atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed).
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; the LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 6 0 11 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

encapsulation pppoe

- Use to configure PPPoE as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation pppoe**

- Use the **no** version to remove PPPoE encapsulation from the interface.

encapsulation vlan

- Use to configure VLAN as the encapsulation method for the interface.
- Example

  host1(config-if)#**encapsulation vlan**

- Use the **no** version to remove VLAN encapsulation from the interface.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 456.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 4/0/2.1**

- Use the **no** version to remove the interface or subinterface.

interface fastEthernet

- Use to select a Fast Ethernet interface.
- Example

  host1(config)#**interface fastEthernet 4/1**

- Use the **no** version to remove IP from an interface or a 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.

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- To specify a Gigabit Ethernet interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format.
- To specify a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
- For more information, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

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

ip address

- Use to assign an IP address and subnet mask to an interface or a subinterface.
- Example

  host1(config-if)#**ip address 192.1.1.1 255.255.255.0**

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

pppoe auto-configure

- Use to set up the router to dynamically create PPPoE subinterfaces on the PPPoE major interfaces.
- Example

  host1(config-subif)#**pppoe auto-configure**

- Use the **no** version to remove this configuration.

pppoe profile

- Use to assign a profile to a static PPPoE major interface. The profile configuration is used to dynamically configure an upper bridged Ethernet, IP, PPP, or PPPoE interface.
- Specify a profile name with up to 80 alphanumeric characters.
- The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.
- Examples

  host1(config-subif)#**pppoe profile pppoeProfile4**
  host1(config-if)#**pppoe profile any anyProfile**

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

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN subinterface.
- Choose one of the following Ethertype values:
  - **8100**—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - **9100**—Specifies Ethertype value 0x9100, which is the default
  - **9200**—Specifies Ethertype value 0x9200
- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-if)#**svlan ethertype 8100**

- Use the **no** version to restore the default value, 9100.

svlan id

- Use to assign S-VLAN IDs and VLAN IDs to VLAN subinterfaces.
- Use S-VLAN ID and VLAN ID numbers that are in the range 0–4095 and that are unique within the Ethernet interface.
- Issue the **svlan id** command before any upper bindings are made, such as IP or PPPoE.

- Example

  host1(config-if)#**svlan id 4 255**

- There is no **no** version.

vlan id

- Use to specify the VLAN ID.
- Use a VLAN ID that is in the range 0–4095 and is unique within the Ethernet interface.
- Issue the **vlan id** command before any upper bindings are made, such as IP or PPPoE.
- Use the optional keyword **untagged** to specify that frames be sent untagged. The keyword is valid only for VLAN ID 0, which can receive tagged frames but sends out untagged frames.
- Example

  host1(config-if)#**vlan id 400**

- There is no **no** version.

Configuring Encapsulation Type Lockout for PPPoE Clients

In configurations with dynamic PPPoE subinterfaces over static PPPoE major interfaces, you can configure dynamic encapsulation type lockout for the PPPoE clients associated with a dynamic PPPoE subinterface column. Using this feature enables you to temporarily prevent the static PPPoE major interface from autodetecting, accepting, and creating dynamic PPPoE subinterface columns for a configurable time period.

By default, encapsulation type lockout is disabled for PPPoE clients. To configure a lockout time range for the PPPoE clients associated with the dynamic PPPoE subinterface columns on the PPPoE major interface, use the **pppoe auto-configure** command with the **lockout-time** keyword. You can also use the **show pppoe interface lockout-time** command to display detailed information about the current lockout condition for each PPPoE client, and the **pppoe clear lockout interface** command to clear (reset) the lockout condition for an individual PPPoE client.

For illustrations of the interface stacking in dynamic PPPoE over static PPPoE configurations, see Figure 42 on page 459, Figure 43 on page 460, Figure 44 on page 461, and Figure 45 on page 462.

Differences from Lockout Configuration for PPPoE over Static ATM

Table 24 lists the important differences between how encapsulation type lockout works for dynamic PPPoE over static PPPoE configurations and how lockout works for dynamic PPPoE over static ATM 1483 configurations.

Table 24: Differences in Lockout Operation for Dynamic PPPoE Configurations

Dynamic PPPoE over Static PPPoE	Dynamic PPPoE over Static ATM 1483
Encapsulation type lockout is disabled by default.	Encapsulation type lockout is enabled by default with a lockout time range of 1–300 seconds.
You must explicitly configure encapsulation type lockout for PPPoE clients with the **pppoe auto-configure** command.	PPPoE clients automatically inherit their lockout setting from the lockout parameters configured for the underlying static ATM 1483 subinterface with the **auto-configure** command. Currently, the dynamic PPPoE interface layer must be configured directly above the static ATM 1483 interface layer to support inheritance of lockout parameters. For an illustration of dynamic PPPoE over static ATM 1483 interface stacking, see Figure 41 on page 453.

For more information about the benefits and operation of dynamic encapsulation type lockout, see *Encapsulation Type Lockout* on page 449. In particular, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451 for information about the rules that apply when you configure the lockout time. These rules are common to both dynamic PPPoE over static PPPoE configurations and dynamic PPPoE over static ATM 1483 configurations.

Configuration Tasks

Configuring dynamic encapsulation type lockout for PPPoE clients includes the following tasks:

- Configuring and verifying lockout for PPPoE clients
- Clearing the lockout condition for a specific PPPoE client

Configuring and Verifying Lockout for PPPoE Clients

To configure and verify encapsulation type lockout for a PPPoE client:

1. Configure the underlying physical interface.

   For example, the following commands configure a static ATM 1483 subinterface and corresponding ATM PVC.

   ```
   host1(config)#interface atm 3/0
   host1(config-if)#interface atm 3/0.101
   host1(config-subif)#atm pvc 10 10 20 aal5snap
   ```

2. Create a static PPPoE major interface.

   ```
   host1(config-subif)#encapsulation pppoe
   ```

3. Configure the PPPoE major interface to detect and accept dynamic PPPoE subinterfaces. Use the **lockout-time** keyword to configure a nondefault lockout time range for the PPPoE clients associated with the dynamic PPPoE subinterface column.

   For example, the following command configures a lockout time in the range 5–60 seconds for the PPPoE clients associated with the dynamic PPPoE subinterface column on the PPPoE major interface.

   host1(config-subif)#**pppoe auto-configure lockout-time 5 60**

4. Assign a profile to the PPPoE major interface.

   host1(config-subif)#**pppoe profile pppoeLockoutProfile**

   For information about creating and using profiles, see *Configuring a Dynamic Interface from a Profile* on page 483.

5. (Optional) Verify the lockout configuration by using either of the following commands.

   - To display summary information about the lockout configuration, use the **show pppoe interface** command. (The following example shows only the portion of the command display relevant to the PPPoE lockout configuration.)

     ```
     host1#show pppoe interface atm 3/0.101
     PPPoE interface ATM 3/0.101 is operStatusUp (dynamic)
     . . .

     Lockout Configuration (seconds): Min 5, Max 60
      Total clients in active lockouts: 0
      Total clients in lockout grace period: 0
     ```

   - To display detailed information about the current lockout condition for each PPPoE client associated with a specific source media access control (MAC) address, use the **show pppoe interface lockout-time** command.

     ```
     host1#show pppoe interface atm 3/0.101 lockout-time
     PPPoE interface ATM 3/0.101
     Lockout Configuration (seconds): Min 5, Max 60
      Total clients in active lockout: 0
      Total clients in lockout grace period: 0
     Client Address Current Elapsed Next
     -------------- ------- ------- ----
     0090.1a10.165e       0       0    5
     ```

   For a description of the fields in the command display, see **show pppoe interface** on page 520 and **show pppoe interface lockout-time** on page 521.

pppoe auto-configure lockout-time

- Use to specify the lockout time range for the PPPoE clients associated with the dynamic PPPoE subinterface column on the static PPPoE major interface.
- Dynamic encapsulation type lockout is disabled for PPPoE clients by default.
- Configuring dynamic encapsulation type lockout temporarily prevents the static PPPoE major interface from detecting, accepting, and creating dynamic PPPoE subinterface columns until the lockout time expires.
- Use the **lockout-time** keyword to set the minimum lockout time and maximum lockout time, each of which can be in the range 1–86400 seconds (24 hours).
- Use the **none** keyword to disable lockout for the PPPoE clients associated with the dynamic PPPoE subinterface column on the static PPPoE major interface.
- For information about the rules that apply when you configure the lockout time for dynamic encapsulation type lockout, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451.
- Example 1—Enables dynamic creation of PPPoE subinterfaces on the static PPPoE major interface using a nondefault lockout time range, 10–120 seconds

  host1(config-subif)#**pppoe auto-configure lockout-time 10 120**

- Example 2—Disables dynamic encapsulation type lockout for any PPPoE clients associated with the dynamic PPPoE subinterface column on the static PPPoE major interface

  host1(config-subif)#**pppoe auto-configure lockout-time none**

- Example 3—Terminates dynamic creation of PPPoE subinterfaces on the static PPPoE major interface and, by extension, disables dynamic encapsulation type lockout for this interface

  host1(config-subif)#**no pppoe auto-configure**

- Use the **no pppoe auto-configure** command to terminate dynamic creation of PPPoE subinterfaces on the static PPPoE major interface.

Clearing the Lockout Condition for a PPPoE Client

You can use the **pppoe clear lockout interface** command to clear the lockout condition for an individual PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface. To identify the PPPoE client, you must specify its source MAC address.

NOTE: Issuing the **pppoe clear lockout interface** command resets the current lockout condition for the specified PPPoE client, but does *not* disable dynamic encapsulation type lockout for that PPPoE client.

To clear the current lockout condition for a PPPoE client:

1. Display the source MAC address assigned to the PPPoE client by issuing one of the following **show** commands:

   - To display the source MAC address when there is no available PPPoE session in progress, use the **show pppoe interface lockout-time** command.

     ```
     host1#show pppoe interface atm 3/0.101 lockout-time
     PPPoE interface ATM 3/0.101
     Lockout Configuration (seconds): Min 5, Max 60
      Total clients in active lockout: 0
      Total clients in lockout grace period: 0
     Client Address Current Elapsed Next
     -------------- ------- ------- ----
     0090.1a10.165e       0       0    5
     ```

   - To display the source MAC address when a subscriber is connected to the router through an available PPPoE session, use either the **show pppoe interface lockout-time** command or the **show pppoe subinterface full** command. (The following example shows only the portion of the command display relevant to the source MAC address.)

     ```
     host1#show pppoe subinterface full
     ...
         PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
     ...
     ```

     For a description of the fields in the command display, see **show pppoe interface lockout-time** on page 521 and **show pppoe subinterface** on page 522.

2. Clear the current lockout condition for the PPPoE client associated with the specified source MAC address on the static PPPoE major interface.

   host1#**pppoe clear lockout interface atm 3/0.101 0090.1a10.165e**

   If the specified PPPoE client is undergoing active lockout or is in a lockout grace period, issuing the **pppoe clear lockout interface** command causes the router to reset the current lockout condition and start the next lockout interval at the minimum configured lockout time.

   The lockout grace period occurs when the time between lockout events is greater than either 15 minutes or the maximum lockout time. When a PPPoE client is in a lockout grace period, the router resets the number of consecutive lockout events to 1. (For more information, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451.)

pppoe clear lockout interface

- Use to clear the lockout condition for the PPPoE client associated with the specified source MAC address.
- For PPPoE clients undergoing active lockout or in a lockout grace period, issuing the **pppoe clear lockout interface** command causes the router to reset the current lockout condition and start the next lockout interval at the minimum configured lockout time.
- You must specify the following:
  - *interfaceType*—One of the following interface types listed in *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*:
    - **atm**
    - **fastEthernet**
    - **gigabitEthernet**
    - **lag**
    - **tenGigabitEthernet**
  - *interfaceSpecifier*—Particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide* for information
  - *macAddress*—Source MAC address of the PPPoE client, specified as a dotted triple of four-digit hexadecimal numbers
- Example

  host1#**pppoe clear lockout interface gigabitEthernet 2/1.1 1011.22c2.333d**

- There is no **no** version.

Configuring IPoA Dynamic Interfaces

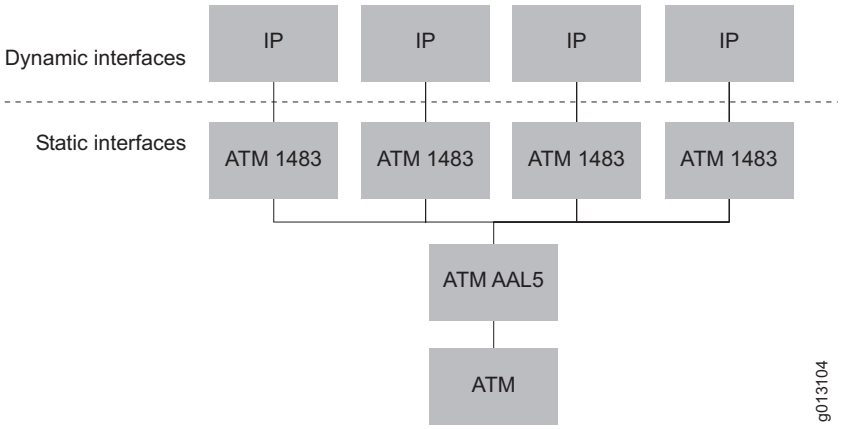
E-series routers support dynamic IP over ATM (IPoA) interfaces. An IPoA interface is IP over ATM 1483 over ATM AAL5 over ATM. See Figure 39 on page 446.

An IPoA configuration is typically used as a high-speed access service or uplink to another router. A common use is to provision IP over ATM circuits over DSL to connect businesses to the Internet—a B-RAS alternative to circuit aggregation. All provisioning can be through the RADIUS server to minimize any configuration of the router.

When IP packets are received over ATM circuits, the IP interfaces are dynamically constructed over the corresponding ATM 1483 interfaces from the configuration data received from the RADIUS server, a profile, or both.

Figure 46 shows the protocol layers that represent the IPoA interface columns, and the layers within the interface columns that are static and dynamic.

Figure 46: Dynamic IPoA over Static ATM 1483 Interface Columns



When you configure dynamic IPoA interfaces, you must assign a profile. Optionally, you can also assign a subscriber identification.

Configuring a Dynamic IPoA Interface

To configure dynamic IPoA interfaces:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM subinterface.

   host1(config-if)#**interface atm 5/0.1**

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**
   host1(config-subif)#**atm pvc 10 100 22 aal5autoconfig**

4. Assign a profile.

   host1(config-subif)#**profile ip foo**

5. (Optional) Assign subscriber identification.

   host1(config-subif)#**subscriber ip user charlie domain myispname password lucy**

6. Do either of the following:

   - Configure the subinterface to detect and accept the dynamic IP encapsulation type using the default lockout time range, 1–300 seconds.

     host1(config-subif)#**auto-configure ip**

   - Configure the subinterface to detect and accept the dynamic IP encapsulation type using a nondefault lockout time range. For example, the following command configures 3600 seconds (1 hour) as the minimum lockout time and 7200 seconds (2 hours) as the maximum lockout time.

     host1(config-subif)#**auto-configure ip lockout-time 3600 7200**

7. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 5/0.1**

atm pvc

- Use to configure a PVC on an ATM interface. Specify one of the following encapsulation types:
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed).
  - **aal5snap**—Specifies an LLC encapsulated circuit; LLC/SNAP header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 6 0 11 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

auto-configure

- Use to configure an ATM subinterface to support a dynamic interface. Specifies the types of dynamic encapsulation that the ATM 1483 subinterface detects and accepts.
- For detailed information about how to use this command, see **auto-configure** on page 455.
- Example 1—Enables autodetection for the IP encapsulation type using the default lockout time range, 1–300 seconds

  host1(config-subif)#**auto-configure ip**

- Example 2—Enables autodetection for the IP encapsulation type using a nondefault lockout time range, 3600–21600 seconds (1–6 hours)

  host1(config-subif)#**auto-configure ip lockout-time 3600 21600**

- Example 3—Disables encapsulation type lockout for the IP encapsulation type

  host1(config-subif)#**auto-configure ip lockout-time none**

- Example 4—Either command reenables encapsulation type lockout for the IP encapsulation type using the default lockout time range

  host1(config-subif)#**auto-configure ip**
  host1(config-subif)#**no auto-configure ip lockout-time**

- Example 5—Permanently locks out the IP encapsulation type until the **auto-configure ip** command is issued

  host1(config-subif)#**no auto-configure ip**

- Use the **no** version to terminate detection of the specified encapsulation type or, if the **lockout-time** keyword is specified, to restore the lockout time range to its default value, 1–300 seconds.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 456.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 4/0/2.1**

- Use the **no** version to remove the interface or subinterface.

profile

- Use to assign a profile.
- You must specify the encapsulation type to which the profile applies: **bridgedEthernet**, **ip**, **ppp**, **pppoe**, or **any**.
- Specify a profile name with up to 80 alphanumeric characters.
- Example

  host1(config-subif)#**profile ppp foo**

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

subscriber

- Use to configure a local subscriber on the E-series router to support authentication and configuration from RADIUS for a dynamic IPoA or bridged Ethernet interface.
- When you configure a subscriber, you must specify the following:
  - *interfaceType*—Type of dynamic interface, **bridgedEthernet** or **ip**
  - *userNameUsage*—How the dynamic interface uses the username for authentication purposes
    - **user**—Use the name as specified.
    - **user-prefix**—Use the name as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the username string. The username format is *userName.slot.port.vpi.vci*. The resulting username string is then used to authenticate with the RADIUS server.
  - *userName*—RADIUS username
  - *domainName*—Domain name
- You may optionally supply password information:
  - *passwordUsage*—How the dynamic interface uses the password for authentication purposes
    - **password**—Use the password as specified.
    - **password-prefix**—Use the password as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the password string. The password format is *password.slot.port.vpi.vci*. The resulting password string is then used to authenticate with the RADIUS server.
  - *password*—RADIUS password
- If your router is running stateful SRP switchover (high availability), the use of the **subscriber** command to configure RADIUS authentication for subscribers on dynamic bridged Ethernet interfaces might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. For more information about using the subscriber management application to bypass this limitation, see *Authenticating Subscribers on Dynamic Bridged Ethernet over Static ATM Interfaces* on page 447.
- Example 1

  host1(config-subif)#**subscriber ip user-prefix charlie domain myisp password-prefix lucy**

- Example 2

  host1(config-subif)#**subscriber bridgedEthernet user westford003 domain acmecorp.east password xyz123**

- Use the **no** version to remove the subscriber.

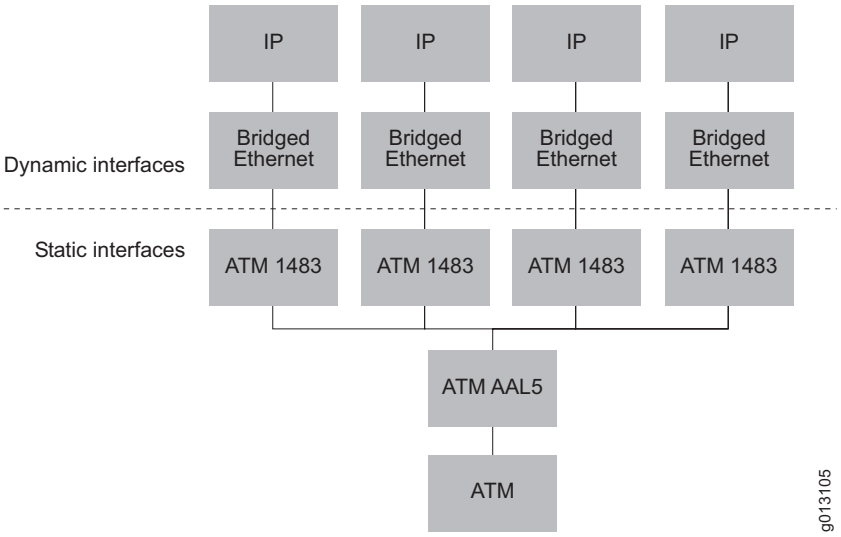
Configuring Bridged Ethernet Dynamic Interfaces

A bridged Ethernet interface is IP over bridged Ethernet over ATM 1483 over ATM AAL5 over ATM.

When bridged Ethernet packets are received over ATM circuits, the bridged Ethernet and IP interfaces are dynamically constructed over the corresponding ATM 1483 interfaces and use the configuration data received from the RADIUS server, a profile, or both.

Figure 47 shows the protocol layers that represent the bridged Ethernet interface columns, and the layers within the interface columns that are static and dynamic.

Figure 47: Dynamic Bridged Ethernet over Static ATM 1483 Interface Columns



Configuring a Dynamic Bridged Ethernet Interface

When you configure dynamic bridged Ethernet interfaces, you must assign a profile. You may optionally assign a subscriber identification.

To configure dynamic bridged Ethernet interfaces:

1. Configure a physical interface.

   host1(config)#**interface atm 5/0**

2. Configure an ATM subinterface.

   host1(config-if)#**interface atm 2/0.1**

3. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   If you want the router to autodetect the encapsulation type, use the **aal5autoconfig** option.

   host1(config-subif)#**atm pvc 10 100 22 aal5snap**
   host1(config-subif)#**atm pvc 10 100 22 aal5autoconfig**

4. Do either of the following:

   - Configure the subinterface to detect and accept the dynamic bridged Ethernet encapsulation type with the default lockout time range, 1–300 seconds.

     host1(config-subif)#**auto-configure bridgedEthernet**

   - Configure the subinterface to detect and accept the dynamic bridged Ethernet encapsulation type with a nondefault lockout time range. For example, the following command configures 3600 seconds (1 hour) as the minimum lockout time and 7200 seconds (2 hours) as the maximum lockout time.

     host1(config-subif)#**auto-configure bridgedEthernet lockout-time 3600 7200**

5. Assign a profile to match the encapsulation type of bridged Ethernet.

   host1(config-subif)#**profile bridgedEthernet foo**

6. (Optional) Assign subscriber identification.

   host1(config-subif)#**subscriber bridgedEthernet user charlie domain myisp password lucy**

7. (Optional) Verify your configuration.

   host1#**show atm subinterface atm 2/0.1**

atm pvc

- Use to configure a PVC on an ATM interface. Select one of the following encapsulation types:
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed).
  - **aal5snap**—Specifies a LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.
- Example

  host1(config-subif)#**atm pvc 6 0 11 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

auto-configure

- Use to configure an ATM subinterface to support a dynamic interface. Specifies the types of dynamic encapsulation that the ATM 1483 subinterface detects and accepts.
- For detailed information about how to use this command, see **auto-configure** on page 455.
- Example 1—Enables autodetection for the bridged Ethernet encapsulation type using the default lockout time range, 1–300 seconds

  host1(config-subif)#**auto-configure bridgedEthernet**

- Example 2—Enables autodetection for the bridged Ethernet encapsulation type using a nondefault lockout time range of 3600–21600 seconds (1–6 hours)

  host1(config-subif)#**auto-configure bridgedEthernet lockout-time 3600 21600**

- Example 3—Disables encapsulation type lockout for the bridged Ethernet encapsulation type

  host1(config-subif)#**auto-configure bridgedEthernet lockout-time none**

- Example 4—Either command reenables encapsulation type lockout for the bridged Ethernet encapsulation type using the default lockout time range

  host1(config-subif)#**auto-configure bridgedEthernet**
  host1(config-subif)#**no auto-configure bridgedEthernet lockout-time**

- Example 5—Permanently locks out the bridged Ethernet encapsulation type until the **auto-configure bridgedEthernet** command is issued

  host1(config-subif)#**no auto-configure bridgedEthernet**

- Use the **no** version to terminate detection of the specified encapsulation type or, if the **lockout-time** keyword is specified, to restore the lockout time range to its default value, 1–300 seconds.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 456.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 4/0/2**

- Use the **no** version to remove the interface or subinterface.

profile

- Use to assign a profile.
- You must specify the encapsulation type to which the profile applies: **bridgedEthernet**, **ip**, **ppp**, **pppoe**, or **any**.
- Specify a profile name with up to 80 alphanumeric characters.
- Example

  host1(config-subif)#**profile bridgedEthernet foo**

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

subscriber

- Use to configure a local subscriber on the E-series router to support authentication and configuration from RADIUS for a dynamic bridged Ethernet or IPoA interface.
- For detailed information about how to use this command, see **subscriber** on page 476.
- Example

  host1(config-subif)#**subscriber bridgedEthernet user-prefix charlie domain myisp password-prefix lucy**

- Use the **no** version to remove the subscriber.

Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces

You can use the JUNOSe subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet over static ATM 1483 interface column, as described in *Authenticating Subscribers on Dynamic Bridged Ethernet over Static ATM Interfaces* on page 447.

To use the subscriber management application to configure IP subscribers on a dynamic bridged Ethernet interface for RADIUS authentication:

1. Define an IP service profile that contains the subscriber's RADIUS authentication parameters including the username, domain, and password.
2. Configure the interface profile from which the router creates a dynamic bridged Ethernet interface column.
   a. Include the desired characteristics for the upper-layer encapsulation types.
   b. (Optional) Specify the name of the route map used to configure the IP subscriber interface.
   c. Use the **bridge1483 service-profile** command to assign the specified IP service profile to the interface profile. The IP service profile contains the RADIUS authentication parameters for subscribers on the dynamic bridged Ethernet interface.

3. Define the underlying static or dynamic ATM 1483 subinterface on which the dynamic bridged Ethernet interface column is built.

    a. Assign the specified interface profile to the ATM 1483 subinterface.

    b. Enable autodetection (autoconfiguration) of the bridged Ethernet upper-layer encapsulation type.

    a. Define the ATM PVC over which data is transmitted.

4. (Optional) Use the **show profile** command to verify assignment of the IP service profile to the interface profile.

    For information, see **show profile** on page 522.

Configuration Example Using subscriber Command

The following configuration example illustrates the preceding procedure. The example has two parts:

- The first part of the example shows how to use the **subscriber** command to configure RADIUS authentication for IP subscribers on a dynamic bridged Ethernet interface. This configuration method *does not support* running stateful SRP switchover on the router.

- The second part of the example shows the commands required to re-create this configuration using the IP subscriber management application. This configuration method uses the **bridge1483 service-profile** command to assign the specified IP service profile to the interface profile, and *does support* running stateful SRP switchover on the router.

Assume that you have issued the following commands to configure IP subscribers on a dynamic bridged Ethernet interface for RADIUS authentication. In this configuration, the **subscriber** command provides the subscriber's authentication parameters, and the static ATM 1483 subinterface is the authenticating layer. Keep in mind that the **subscriber** command does not support running stateful SRP switchover on the router.

```
! Configure the interface profile from which to create a dynamic bridged Ethernet
! interface. Include the desired attributes (in this case, IGMP) and, optionally, the
! name of the route map used to configure the IP subscriber interface.
host1(config)#profile east
host1(config-profile)#ip igmp
host1(config-profile)#ip igmp immediate-leave
host1(config-profile)#ip igmp group limit 6
host1(config-profile)#ip route-map ip-subscriber eastRouteMap
host1(config-profile)#exit
!
! Configure the static ATM 1483 subinterface to assign the east profile, support
! RADIUS authentication, enable autodetection of the bridged Ethernet upper-layer
! encapsulation type, and define the ATM PVC.
host1(config)#interface atm 2/1.100 point-to-point
host1(config-subif)#profile bridgedEthernet east
host1(config-subif)#subscriber bridgedEthernet user westford001
domain xyzcorp.east password abc123
host1(config-subif)#auto-configure bridgedEthernet
```

```
host1(config-subif)#atm pvc 100 10 101 aal5snap 6400 0 0
host1(config-subif)#exit
```

Equivalent Configuration Example Using IP Subscriber Management

To achieve the same functionality without adversely affecting stateful SRP switchover if it is running on the router, you can issue the following commands to use the subscriber management feature to configure IP subscribers on a dynamic bridged Ethernet interface using RADIUS. In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application is the authenticating layer. To assign the IP service profile to the interface profile, use the **bridge1483 service-profile** command.

```
! Define an IP service profile containing the subscriber's username, domain,
and password.
host1(config)#ip service-profile eastServiceProfile
host1(config-service-profile)#user-name westford001
host1(config-service-profile)#domain xyzcorp.east
host1(config-service-profile)#password abc123
host1(config-service-profile)#exit
!
! Configure the interface profile from which to create a dynamic bridged Ethernet
! interface. Include the desired attributes (in this case, IGMP), the name of the
! route map used to configure the IP subscriber interface (optional), and the name
! of the IP service profile containing the authentication parameters for the dynamic
! bridged Ethernet interface.
host1(config)#profile east
host1(config-profile)#ip igmp
host1(config-profile)#ip igmp immediate-leave
host1(config-profile)#ip igmp group limit 6
host1(config-profile)#ip route-map ip-subscriber eastRouteMap
host1(config-profile)#bridge1483 service-profile eastServiceProfile
host1(config-profile)#exit
!
! Configure the static ATM 1483 subinterface to assign the east profile,
! enable autodetection of the bridged Ethernet upper-layer encapsulation type,
! and define the ATM PVC.
host1(config)#interface atm 2/1.100 point-to-point
host1(config-subif)#profile bridgedEthernet east
host1(config-subif)#auto-configure bridgedEthernet
host1(config-subif)#atm pvc 100 10 101 aal5snap 6400 0 0
host1(config-subif)#exit
```

For more information about using the subscriber management application, see *JUNOSe Broadband Access Configuration Guide, Chapter 23, Configuring Subscriber Management*.

bridge1483 service-profile

- Use from Profile Configuration mode to assign the specified IP service profile to the interface profile from which a dynamic bridged Ethernet interface is created.
- The IP service profile must be defined in the default virtual router.
- Example

  host1(config-profile)#**bridge1483 service-profile westServiceProfile**

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

Configuring a Dynamic Interface from a Profile

You define profiles by using CLI commands similar to the ones you use to configure static interfaces. When configuring profiles, you can specify every layer explicitly or specify a subset of layers.

Profile Considerations

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

In contrast to static PPP interfaces (above which only dynamic IP interfaces can be created), static ATM 1483 subinterfaces support recognition and creation of the following upper dynamic interface types or *encapsulations*: bridged Ethernet, IP, IPv6, Multilink PPP, PPP, and PPPoE interfaces. The **auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **auto-configure** command.

In contrast to dynamic ATM 1483 subinterfaces, dynamic VLAN subinterfaces support recognition and creation of simultaneous IP and PPPoE upper dynamic interface types. The **vlan auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure a VLAN subinterface with distinct profile assignments for each encapsulation type supported by the **vlan auto-configure** command.

Each profile typically contains configuration attributes for the expected encapsulation, in addition to attributes for other higher-interface layers through IP. If your configuration of upper layers is intended to be different depending on which incoming encapsulation is received by the subinterface, configure and assign separate profiles for each encapsulation type. If your configuration of upper layers is the same for more than one encapsulation type, configure one profile and assign it for those encapsulation types.

Profile Characteristics

Currently, profiles support bridged Ethernet, IP, IPv6, L2TP, Multilink PPP, PPP, PPPoE, and VLANs. You create a profile with a specific set of characteristics. You then assign the profile to multiple interfaces instead of creating separate interfaces with identical attributes. After you create a profile, you can assign it to static ATM 1483, static PPP, or static VLAN major interfaces on different devices.

Bridged Ethernet Characteristics

A profile can contain the following bridged Ethernet characteristic:

- mtu—Sets the maximum allowable size, in bytes, of the maximum transmission unit (MTU) for dynamic bridged Ethernet interfaces

IP Characteristics

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

- access-routes—Enables the creation of host access routes on an interface
- address—Configures an IP address on an interface
- auto-configure ip-subscriber—Configures a primary IP interface to enable dynamic creation of subscriber interfaces
- auto-detect ip-subscriber—Enables packet detection on the router and specifies that IP automatically detects packets that do not match any entries in the demultiplexer table
- directed-broadcast—Enables directed broadcast forwarding
- filter-options all—Filters out packets that include IP options
- igmp—Configures an IGMP interface
- ignore-df-bit—Specifies that the don't-fragment bit is ignored
- inactivity-timer—Configures an inactivity timer value for IP interfaces
- inspection—Associates an inspection list to the interface for firewalling
- mtu—Configures the MTU 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-cache flow sampled—Enables J-Flow statistics on an interface
- route-map ip-subscriber—Configures the interface for route-map processing
- sa-validate—Verifies that a packet has been sent from a valid source address

- tcp adjust-mss—Modifies maximum segment size (MSS) 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 (VR) to which interfaces created by this profile attach

IPv6 Characteristics

A profile can contain one or more of the following IPv6 characteristics:

- address—Configures an IPv6 address on an interface
- nd—Enables Neighbor Discovery on an interface
- nd managed-config-flag—Sets the "managed address configuration" flag in IPv6 router advertisements
- nd other-config-flag—Sets the "other stateful configuration" flag in IPv6 router advertisements
- nd prefix-advertisement—Specifies which IPv6 prefixes are included in IPv6 router advertisements
- nd ra-interval—Configures the interval between IPv6 router advertisements
- nd ra-lifetime—Configures the router advertisement lifetime
- nd reachable-time—Configures the amount of time the router can reach an IPv6 node after a reachability confirmation event occurs
- nd suppress-ra—Disables router advertisement transmissions
- mld—Configures the multicast listener discovery (MLD) interface
- mtu—Configures the MTU for a network
- 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 attach

L2TP Characteristics

A profile can contain the following L2TP characteristic:

- policy—Assigns an L2TP policy list to a profile

MLPPP and PPP Characteristics

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

- aaa-profile—Assigns an AAA profile
- authentication—Requests PAP or CHAP authentication from a PPP peer
- authentication virtual router—Specifies a virtual router for the authentication virtual router context
- chap challenge length—Modifies the length of the CHAP challenge
- fragmentation—Enables fragmentation on an MLPPP link interface
- hash-link-selection—Enables use of a hash-based algorithm to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on dynamic MLPPP interfaces
- initiate-ip—Initiates IPv4 for passive clients
- initiate-ipv6—Initiates IPv6 for passive clients
- ipcp netmask—Controls the negotiation of the IPCP netmask option 0x90; *disabled* indicates do not negotiate, *enabled* indicates negotiate
- keepalive—Specifies a keepalive value, in seconds
- log—Enables packet or state machine logging for any dynamic interfaces that use the profile
- magic-number disable—Disables negotiation of the local magic number
- magic-number ignore-mismatch—Causes the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.
- mru—Configures the maximum receive unit size for the interface
- multilink enable—For MLPPP interfaces only, enables the creation of dynamic MLPPP interfaces
- passive-mode—Forces the interface into passive mode before LCP negotiation begins, for a period of one second to enable slow clients to start up and initiate the LCP negotiation
- peer dns—Resolves conflicts when the E-series router and the PPP peer system have the primary and secondary DNS addresses configured with different values

- peer wins—Resolves conflicts when the E-series router and the PPP peer system have the primary and secondary WINS addresses configured with different values
- reassembly—Enables reassembly on an MLPPP link interface

PPPoE Characteristics

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

- AC name—Adds an access concentrator name to the profile configuration
- always-offer—Causes the router to offer to set up a session for the client, even when the router has insufficient resources to establish a session
- duplicate-protection—Prevents a client from establishing more than one session using the same MAC address
- log pppoeControlPacket—Enables packet trace logging on PPPoE dynamic interfaces created with this profile
- motm—Causes the router to send a PPPoE Active Discovery Message (PADM) message of the minute
- mtu—Configures the MTU
- remote-circuit-id—Enables the router to capture and process a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM) device
- service-name-table—Assigns a PPPoE service name table to dynamic interfaces created with this profile
- sessions—Specifies the maximum number of subinterfaces permitted on a PPPoE major interface
- url—Causes the PPPoE application to send a URL string to the new client

VLAN Characteristics

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

- advisory-rx-speed—Sets an advisory receive speed for VLAN subinterfaces
- advisory-tx-speed—Sets an advisory connect speed for VLAN subinterfaces
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic VLAN subinterface
- auto-configure agent-circuit-identifier—Enables the creation of VLAN subinterfaces that are based on agent-circuit-identifier information
- description—Assigns a description to VLAN subinterfaces that are created with this profile
- policy—Attaches (or removes) a policy to (or from) a dynamically created VLAN

- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the VLAN subinterface
- service-profile—Specifies a service profile name to a dynamically created VLAN
- svlan ethertype—Specifies that the packet must use this Ethertype to create the dynamic VLAN subinterface

Working with Profiles

Figure 48 shows how to create a profile and assign characteristics to it.

Figure 48: Creating and Configuring a Profile

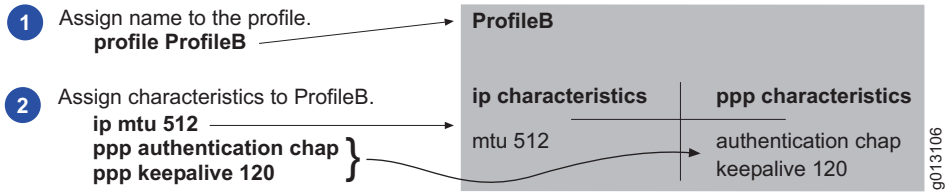
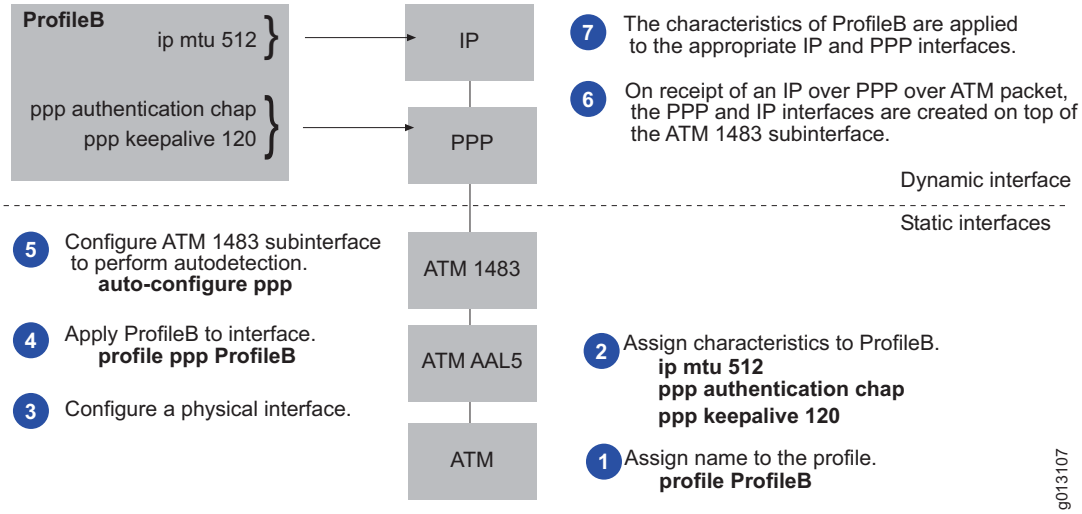


Figure 49 shows how to assign a profile to static interfaces. These static interfaces create dynamic interfaces above them.

Figure 49: Assigning a Profile to a Static Interface



Configuring a Profile

You can create a profile by using CLI commands similar to those used to create the equivalent static interfaces. You can configure a profile for bridged Ethernet, IP, IPv6, MLPPP, PPP, PPPoE, or VLAN interfaces.

To configure a profile:

1. Create a profile by assigning it a name.

   host1(config)#**profile foo**

2. Specify a VR to which to assign dynamic IP interfaces created with this profile.

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

3. Specify an IP loopback interface for dynamic IP interfaces created with this profile to be associated.

   host1(config-profile)#**ip unnumbered loopback 0**

4. Configure IPCP option 0x90.

   host1(config-profile)#**ppp ipcp netmask**

5. Optionally set IP, IPv6, MLPPP, PPP, or PPPoE characteristics.

NOTE: When configuring either IP or IPv6 to operate over PPP, you might want to initiate IP or IPv6 by using the appropriate **ppp initiate** command, either **ppp initiate-ip** or **ppp initiate-ipv6**. This command initiates either IPv4 or IPv6 in the event you are connecting to a passive client.

bridge1483 mtu

- Use to set the maximum allowable size, in bytes, of the MTU for bridged Ethernet interfaces.
- Specify an MTU size in the range 64–9180 bytes.
- Example

  host1(config-profile)#**bridge1483 mtu 1684**

- Use the **no** version to restore the default MTU size for bridged Ethernet interfaces, 1518 bytes.

ip access-routes

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

  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.
- Example

  host1(config-profile)#**ip address 192.13.5.61**

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

ip auto-configure ip-subscriber

- Use to configure a primary IP interface to enable dynamic creation of subscriber interfaces.
- Use the **include-primary** keyword to specify that the primary interface is assigned to the first subscriber.
- Use the **exclude-primary** keyword to specify that the primary interface is not used for dynamic subscribers. By default, the primary interface is not assigned to a dynamic subscriber.
- Example

  host1(config-profile)#**ip auto-configure ip-subscriber include-primary**

- Use the **no** version to disable creation of dynamic subscriber interfaces associated with this primary IP interface. Use the **no** version with the **include-primary** keyword to specify that the primary interface is not assigned to a subscriber. Use the **no** version with the **exclude-primary** keyword to specify that the primary interface is assigned to a subscriber.

ip auto-detect ip-subscriber

- Use to enable packet detection on the router and specify that IP automatically detect packets that do not match any entries in the demultiplexer table.
- Example

  host1(config-profile)#**ip auto-detect ip-subscriber**

- Use the **no** version to restore the default behavior, which disables packet detection.

ip directed-broadcast

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

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

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

ip filter-options all

- Use to filter out packets that include IP options.
- Example

  host1(config-profile)#**ip filter-options all**

- Use the **no** version to disable filtering of packets with IP options.

ip igmp

- Use to enable IGMP on an interface, and sets the IGMP version to IGMPv2.
- Example

  host1(config-profile)#**ip igmp**

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

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.

NOTE: You can also use RADIUS VSA [26-70] to configure the router's DF bit support. The action configured by the RADIUS VSA takes precedence over the action configured by the **ip ignore-df-bit** command. For more information, see *JUNOSe Broadband Access Configuration Guide, Chapter 3, Configuring RADIUS Attributes*.

- Example

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

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

ip inactivity-timer

- Use to configure an inactivity timer value for an IP interface.
- Example

  host1(config-profile)#**ip inactivity-timer 100**

- Use the **no** version to restore the default behavior, which disables the inactivity timer.

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 maximum transmission unit size sent on an IP interface.
- Example

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

- Use the **no** version to restore the default value, 0, which means that the router takes the value from a lower protocol layer.

ip nat

- Use to mark interfaces that participate in NAT translation as residing on the inside or the outside network.
- Example

  host1(config-profile)#**ip nat inside**

- Use the **no** version to unmark the interface (the default) so that it does not participate in NAT translation.

ip policy

- Use to assign a policy list to the ingress or egress of an interface to which the profile is attached.
- Example

  host1(config-profile)#**ip policy secondary-input my-policy**

- Use the **no** version to remove the association between a policy list and a 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-profile)#**ip redirects**

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

ip route-cache flow sampled

- Use to enable J-Flow statistics on the interface.
- Example

  host1(config-profile)#**ip route-cache flow sampled**

- Use the **no** version to delete J-Flow statistics from the profile.

ip route-map ip-subscriber

- Use to configure an interface for route-map processing and specify the route map that is applied to the IP interface subscriber.
- Example

  host1(config-profile)#**ip route-map ip-subscriber chicagoRouteMap**

- Use the **no** version to delete the route map.

ip sa-validate

- Use to enable source address validation on an IP interface.
- Source address validation verifies that a packet has been sent from a valid source address.

- Example

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

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

ip tcp adjust-mss

- Use to modify the maximum segment size (MSS) for TCP SYN packets traveling through the interface.
- Example

  host1(config-profile)#**ip tcp adjust-mss 200**

- Use the **no** version to remove the MSS modification.

ip unnumbered

- Use to specify the unnumbered interface with which dynamic interfaces created with the profile are associated.
- You can configure a loopback using RADIUS instead of adding one to the profile using the **ip unnumbered loopback** command.
- Example

  host1(config-profile)#**ip unnumbered loopback 5**

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

ip virtual-router

- Use to assign a virtual router (VR) to a profile. Interfaces created by the profile are attached to this VR.
- If the VR specified in a profile with the **ip virtual-router** command differs from the VR provided by AAA, IP uses the VR provided by AAA when the dynamic IP upper-layer interface is created. For more information about using the **ppp authentication virtual-router** command, see **ppp authentication** on page 497.
- Example

  host1(config-profile)#**ip virtual-router salem1**

- Use the **no** version to remove the VR assignment from the profile. If no VR is specified via RADIUS, then any subsequent use of the profile to create a dynamic interface fails for lack of a VR.

ipv6 address

- Use to configure an IPv6 address on an interface to which the profile is attached.
- Example

  host1(config-profile)#**ipv6 address 1::1/64**

- Use the **no** version to remove the IPv6 address from the interface.

ipv6 mld

- Use to enable MLD on an interface, and set the MLD version to MLDv2.
- Example

  host1(config-profile)#**ipv6 mld**

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

ipv6 mtu

- Use to set the maximum transmission unit size of IPv6 packets sent on an interface.
- Example

  host1(config-profile)#**ipv6 mtu 1000**

- Use the **no** version to restore the default value, 0, which means that the router takes the value from a lower protocol layer.

ipv6 nd

- Use to enable the IPv6 Neighbor Discovery process on an interface.
- Example

  host1(config-profile)#**ipv6 nd**

- Use the **no** version to disable the Neighbor Discovery process.

ipv6 nd managed-config-flag

- Use to set the "managed address configuration" flag in IPv6 router advertisements.
- Example

  host1(config-profile)#**ipv6 nd managed-config-flag**

- Use the **no** version to clear the flag from IPv6 router advertisements.

ipv6 nd other-config-flag

- Use to set the "other stateful configuration" flag in IPv6 router advertisements.
- Example

  host1(config-profile)#**ipv6 nd other-config-flag**

- Use the **no** version 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-profile)#**ipv6 nd prefix-advertisement 2002:1::/64 60000 45000 onlink autoconfig**

- Use the **no** version 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-profile)#**ipv6 nd ra-interval 500**

- Use the **no** version 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-profile)#**ipv6 nd ra-lifetime 900**

- Use the **no** version to restore the default lifetime, 1800 seconds.

ipv6 nd reachable-time

- Use to specify the amount of time, in milliseconds, that the E-series router can reach a remote IPv6 node after some reachability confirmation event has occurred.
- Example—Sets the reachable-time to 30,000 milliseconds

  host1(config-profile)#**ipv6 nd reachable-time 30000**

- Use the **no** version to restore the default value 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 LAN local area network (Ethernet) interface.
- Example

  host1(config-profile)#**ipv6 nd suppress-ra**

- Use the **no** version to reenable the sending of IPv6 router advertisement transmissions on the LAN (Ethernet) interface

ipv6 policy

- Use to assign a policy list to the ingress or egress of an interface to which the profile is attached.
- Example

  host1(config-profile)#**ipv6 policy secondary-input my-policy**

- Use the **no** version to remove the association between a policy list and a profile.

ipv6 sa-validate

- Use to enable source address validation on an IPv6 interface.
- Source address validation verifies that a packet has been sent from a valid source address.
- Example

  host1(config-profile)#**ipv6 sa-validate**

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

ipv6 unnumbered

- Use to enable or disable IPv6 processing on an interface without assigning an explicit IPv6 address to that interface.
- Example

  host1(config-profile)#**ipv6 unnumbered loopback 0**

- Use the **no** version to remove the IPv6 address from the interface.

ipv6 virtual-router

- Use to specify a VR in an IPv6 profile. Dynamic interfaces created with the profile are assigned to this VR.
- Example

  host1(config-profile)#**ipv6 virtual-router westford01**

- Use the **no** version to remove the VR assignment from the profile. If no VR is specified via RADIUS, then any subsequent use of the profile to create a dynamic interface fails for lack of a VR.

l2tp policy

- Use to assign a policy list to the ingress or egress of an interface to which the profile is attached.
- Example

  host1(config-profile)#**l2tp policy secondary-input my-policy**

- Use the **no** version to remove the association between a policy list and a profile.

ppp aaa-profile

- Use to assign an AAA profile to static and dynamic, multilink and nonmultilink PPP interfaces.
- The PPP application associates the AAA profile with the interface and passes the AAA profile to AAA for authentication.
- If an AAA profile is deleted after it has been assigned to an interface, AAA denies the authentication and logs a message.

- When you remove an AAA profile, it does not remove any corresponding bindings between PPP interfaces or interface profiles and the AAA profile. If an AAA profile with the same name is added, the interface cannot authenticate until the AAA profile is reassigned.

NOTE: Although an AAA profile and an interface profile have similar functionality, they are not related and you need to treat them differently.

- Example

  host1(config-profile)#**ppp aaa-profile westford24**

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

NOTE: For more information about AAA profiles, see *JUNOSe Broadband Access Configuration Guide, Chapter 1, Configuring Remote Access*.

ppp authentication

- Use to require authentication from the PPP peer.
- To specify the name of a virtual router (VR) to be used as the authentication VR context, use the **virtual-router** keyword. Keep the following points in mind when you use the **ppp authentication virtual-router** command:
  - When you specify a VR in the **ppp authentication** command, AAA does not query the domain map for the assigned VR context. Instead, AAA uses the VR specified in the **ppp authentication** command as the authentication VR context and issues the authentication request to the authentication server in the assigned VR context.
  - If you specify the default VR as the authentication VR context, AAA loosely binds the user to the default VR. This means that RADIUS *can override* the default VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies the default VR, AAA returns either the default VR or the VR specified by RADIUS.
  - If you specify a VR other than the default VR as the authentication VR, AAA tightly binds the user to the specified VR. This means that RADIUS *cannot override* the specified VR context with a new VR context during the authentication process. When the **ppp authentication virtual-router** command specifies a nondefault VR, AAA returns the specified VR.
  - If the VR specified in a profile with the **ip virtual-router** command differs from the VR provided by AAA, IP uses the VR provided by AAA when the dynamic IP upper-layer interface is created. For more information about using the **ip virtual-router** command, see **ip virtual-router** on page 493.
- The router supports the MD5 authentication algorithm for CHAP authentication.

- Example 1—Specifies PAP or CHAP as the primary authentication protocol, and the other authentication protocol as the alternative. For example, the following command specifies **pap** as the primary authentication protocol and **chap** as the alternate.

  host1(config-if)#**ppp authentication pap chap**

  The router requests the use of PAP as the authentication protocol (because it appears first in the command line). If the peer refuses to use PAP, the router requests the CHAP protocol. If the peer refuses to negotiate authentication, the router terminates the PPP session.

NOTE: The JUNOSe software's PPP application accepts null usernames during PAP and CHAP authentication. When the PPP application receives an authentication request that includes a null username, PPP passes the request to AAA. To take advantage of this feature, configure your authentication server to support the use of null usernames.

- Example 2—Specifies a virtual router for the authentication virtual router context. This command is available in static configurations and in profiles.

  host1(config-if)#**ppp authentication virtual-router boston pap chap**

- Use the **no** version to specify that the router does not require authentication.

ppp chap-challenge-length

- Use to modify the length of the CHAP challenge by specifying the minimum length and maximum length.

CAUTION: Do *not* use the **ppp chap-challenge-length** command; increasing the minimum length (from the default 16 bytes) or decreasing the maximum length (from the default 32 bytes) reduces the security of your router.

- Specify the minimum and maximum lengths in bytes in the range 8–63.
- The maximum length must be greater than or equal to the minimum length.
- Example

  host1(config-profile)#**ppp chap-challenge-length 24 28**

- Use the **no** version to restore the default minimum 16 bytes and default maximum 32 bytes.

ppp fragmentation

- Use to enable fragmentation on an MLPPP link interface and optionally specify the maximum fragment size, in octets, to be used on the link.
- Example

  host1(config-profile)#**ppp fragmentation 128**

- Use the **no** version to disable fragmentation on the link and restore the default fragment size, which is the link's MTU.

ppp hash-link-selection

- Use to enable use of a hash-based algorithm to select the link on which the router transmits non-best-effort (high-priority) packets, such as voice or video, on the dynamic MLPPP interfaces created by this profile.
- Hash-based MLPPP link selection is available only for non-best-effort traffic. For best-effort traffic, the router uses a round-robin algorithm for link selection.
- Using hash-based link selection instead of the default round-robin link selection for non-best-effort traffic ensures that the router maintains the proper packet order when transmitting high-priority packets.
- When you configure hash-based link selection, the router uses the IP source address and IP destination address of the packet as a hash to select the MLPPP member link on which to transmit the packet.
- Example—The following commands configure hash-based MLPPP link selection for all dynamic MLPPP interfaces created by the profile named dynamicMlppp.

  host1(config)#**profile dynamicMlppp**
  host1(config-profile)#**ppp multilink enable**
  host1(config-profile)#**ppp hash-link-selection**

- Use the **no** version to restore the default round-robin algorithm for MLPPP link selection.

ppp initiate-ip

- Use to initiate IPv4 for passive clients. By default, PPP creates IP instances when it receives client requests.
- Example

  host1(config-profile)#**ppp initiate-ip**

- Use the **no** version to disable initiation of IP.

ppp initiate-ipv6

- Use to initiate IPv6 for passive clients. By default, PPP creates IPv6 instances when it receives client requests.
- Example

  host1(config-profile)#**ppp initiate-ipv6**

- Use the **no** version to disable initiation of IPv6.

ppp ipcp netmask

- Use to specify Internet Protocol Control Protocol (IPCP) option 0x90 for each PPP interface. By default, IPCP option 0x90 is disabled on the interface.
- Example

  host1(config-profile)#**ppp ipcp netmask**

- Use the **no** version to disable IPCP option 0x90 option on the interface.

ppp keepalive

- Use to specify the keepalive timeout value.
- This command always operates in high-density keepalive mode when PPP is layered over ATM or PPPoE.
- When the keepalive timer expires, the interface searches for frames received from the peer in the prior keepalive timeout seconds. If the interface finds such frames, it does not send an LCP echo request (keepalive). Keepalive packets are sent only if the peer is silent (no traffic was received from the peer during the previous keepalive timeout interval). If both sides are configured with keepalive, receipt of an LCP echo request by one end suppresses the transmission of an LCP echo request by that end.
- You can specify a timeout value in the range 30–64800 seconds. The default value is 30 seconds.
- If the keepalive interval is 30 seconds, a failed link is detected between 90 and 120 seconds after failure.
- Use **ppp keepalive** without a value to restore the default, 30 seconds.
- Example

  host1(config-profile)#**ppp keepalive 50**

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

ppp log

- Use to enable PPP packet or state machine logging on any dynamic interface that uses the profile being configured. Specify one of the following keywords:
  - **pppPacket**—Enables PPP packet logging
  - **pppStateMachine**—Enables PPP state machine logging
- Example

  host1(config-profile)#**ppp log pppPacket**

NOTE: This command is equivalent to the l**og severity debug pppPacket** and **log severity debug pppStateMachine** commands.

- Use the **no** version to disable packet or state machine logging.

ppp magic-number disable

- Use to disable negotiation of the local magic number.
- Issuing this command prevents the router from detecting loopback configurations.
- Example

  host1(config-profile)#**ppp magic-number disable**

- Use the **no** version to restore negotiation of the local magic number.

ppp magic-number ignore-mismatch

- Use to cause the router to ignore a mismatch of the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number.
- For more information about using this command, see *Validation of LCP Peer Magic Number* in *Chapter 7, Configuring Point-to-Point Protocol*.
- To verify configuration of LCP peer magic number validation on the router, use the **show profile** command. For information, see **show profile** on page 522.
- Example

  host1(config-if)#**ppp magic-number ignore-mismatch**

- Use the **no** version to restore the default behavior, in which the router terminates the PPP connection if it detects an LCP peer magic number mismatch.

ppp mru

- Use to control the negotiation of the maximum receive unit (MRU).
- Specify the number of bytes, in the range 64–65535.
- We recommend you coordinate this value with the network administrator on the other end of the line.
- If the value configured for the PPP MRU is greater than the value of the lower-layer MRU minus the PPP header length, the router logs a warning message and uses the lesser of the configured MRU value or the lower-layer MRU value minus the PPP header length to negotiate the local MRU.
- If the value configured for the PPP MRU conflicts with a similar value configured for another protocol, such as the MTU value for PPPoE, the router uses the lesser of the two values.
- Example

  host1(config-if)#**ppp mru 576**

- Use the **no** version to restore the default value, which causes PPP to use the lower-layer MRU minus the PPP header length as the MRU value.

ppp multilink enable

- Use in a profile to enable the creation of dynamic MLPPP interfaces.
- Example

  host1(config-profile)#**ppp multilink enable**

- Use the **no** version to cause the LNS to reject any incoming requests to create dynamic MLPPP interfaces.

ppp passive-mode

- Use to force a static or dynamic PPP interface into passive mode before LCP negotiation begins, for a period of one second. This delay enables slow clients to start up and initiate the LCP negotiation.
- Example

  host1(config-profile)#**ppp passive-mode**

- Use the **no** version to disable passive mode.

ppp peer

- Use to resolve conflicts when the router and the PPP peer system have the primary and secondary DNS and WINS addresses configured with different values.
- By default, the DNS and WINS addresses configured on the router take precedence.
- Use the **ppp peer dns** command or the **ppp peer wins** command to configure the PPP peer system as the one that takes precedence. The **ppp peer** command has no effect unless both systems have the address configured and the address is in conflict. If the PPP peer system has the address and the router does not, the peer always supplies the address regardless of how you have configured the PPP peer.
- Example

  host1(config-profile)#**ppp peer dns**

- Use the **no ppp peer dns** command or the **no ppp peer wins** command when you want the router to take precedence during setup negotiations between the router and the remote PC client. If the IP addresses passed to the router by the remote PC client differ from the ones you have configured on your router, the router returns the values that you configured as the correct values to the remote PC client.

ppp reassembly

- Use to enable reassembly on an MLPPP link interface and optionally specify the administrative MRRU value, in octets, for the link.
- Example

  host1(config-profile)#**ppp reassembly 1590**

- Use the **no** version to disable reassembly on the link and restore the default value, which is the link's local MRU.

pppoe acName

- Use to add an access concentrator (AC) name to the profile configuration.
- Example

  host1(config-profile)#**pppoe acName CYM9876**

- Use the **no** version to remove the AC name.

pppoe always-offer

- Use to set up the router to offer to set up a session for the client, even if the router has insufficient resources to establish a session.
- This feature is disabled by default.
- Example

  host1(config-profile)#**pppoe always-offer**

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

pppoe duplicate-protection

- Use to prevent a client from establishing more than one session using the same MAC address.
- This feature is disabled by default.
- Example

  host1(config-profile)#**pppoe duplicate-protection**

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

pppoe log pppoeControlPacket

- Use to enable packet trace logging on PPPoE dynamic interfaces created with this profile. Packet trace information is logged to the pppoeControlPacket log.
- Example

  host1(config-profile)#**pppoe log pppoeControlPacket**

- Use the **no** version to turn off packet trace logging.

pppoe motm

- Use to cause the PPPoE application to send the string to the new client created when the profile is dynamically attached to an IP interface.
- The message string is saved in nonvolatile storage (NVS).
- Example

  host1(config-profile)#**pppoe motm string**

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

pppoe mtu

- Use to set the MTU using a combination of lower layer restrictions and controls.
- You can specify an MTU greater than the current maximum permitted by RFC 2516, in the range 66–65535.
- You can use the **use-lower-layer** keyword to use the lower layer interface value minus any PPPoE overhead. You can use the **use-mtu-tag** keyword to use the provided PPPoE mtu tag value.

- Example

  host1(config-profile)#**pppoe mtu 1380**

- Use the **no** version to restore the default value, 1494.

pppoe remote-circuit-id

- Use to enable the router to capture and process a vendor-specific tag containing a remote circuit ID transmitted from a DSLAM device.
- Optionally, the router can use the remote circuit ID in place of either or both of the Calling-Station-Id [31] and NAS-Port-Id [87] RADIUS attributes to uniquely identify subscriber locations.
- Example

  host1(config-profile)#**pppoe remote-circuit-id**

- Use the **no** version to restore the default behavior, which is not to capture and process the remote circuit ID.

pppoe service-name-table

- Use to assign a PPPoE service name table to dynamic interfaces created with this profile.
- A PPPoE service name table defines the set of specific service name tags that an AC, such as an E-series router, offers to PPPoE clients. It also controls whether the router responds to or does not respond to client requests containing an empty service name tag.
- Specify the name of the PPPoE service name table configured with the **pppoe-service-name-table** command from Global Configuration mode.
- Example

  host1(config-profile)#**pppoe service-name-table myServiceTable1**

- Use the **no** version to remove the PPPoE service name table assignment.

pppoe sessions

- Use to specify the maximum number of PPPoE subinterfaces permitted on an interface, in the range 1–8000 (ERX routers) or 1–16,000 (E120 and E320 routers). The default value is 8000 (ERX routers) or 16,000 (E120 and E320 routers).
- The **sessions** command affects only the creation of subinterfaces after the command is entered. Previously created interfaces remain, even if their number exceeds the new value of the **sessions** parameter.
- Example

  host1(config-profile)#**pppoe sessions 3000**

- Use the **no** version to restore the default value, 8000 (ERX routers) or 16,000 (E120 and E320 routers).

pppoe url

- Use in a profile to cause the PPPoE application to send the string to the new client created when the profile is dynamically attached to an IP interface.
- The message string is saved in nonvolatile storage (NVS).
- PPPoE substitutes certain characters for information in the specified URL string before transmitting:

  %U username and domain name

  %u username

  %d domain name

  %D profile name

  %% % character
- Example

  host1(config-profile)#**pppoe url http://www.relevanturl.com**
- Use the **no** version to disable the command.

profile

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

  host1(config)#**profile foo**
- Use the **no** version to remove a profile.

svlan ethertype

- Use to assign an Ethertype value for the S-VLAN subinterface in a profile.
- Choose one of the following Ethertype values:
  - **8100**—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - **88a8**—Specifies Ethertype value 0x88a8, as defined in draft IEEE Standard 802.1ad
  - **9100**—Specifies Ethertype value 0x9100, which is the default
- Use an Ethertype value that matches the Ethertype value set on the customer premises equipment (CPE) to which your router connects.
- Example

  host1(config-profile)#**svlan ethertype 8100**
- Use the **no** version to restore the default value, 9100.

vlan advisory-rx-speed

- Use to set an advisory receive speed for VLAN subinterfaces that are created with the profile you are configuring. For detailed information about how to use this command, see **vlan advisory-rx-speed** on page 588.
- Example

  host1(config-profile)#**vlan advisory-rx-speed 2000**

- Use the **no** version to restore the default behavior—the Rx speed is not sent to the LNS.

vlan advisory-tx-speed

- Use to set an advisory connect speed for VLAN subinterfaces that are created with the profile that you are configuring.For detailed information about how to use this command, see **vlan advisory-tx-speed** on page 588.
- Example

  host1(config-profile)#**vlan advisory-tx-speed 2000**

- Use the **no** version to restore the default behavior—the Tx speed is not sent to the LNS.

vlan auto-configure

- Use to specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic VLAN subinterface.
- Include this command in the base profile for a dynamic VLAN subinterface.
- Example

  host1(config-profile)#**vlan auto-configure ip**

- Use the **no** version to terminate detection of the specified encapsulation type.

vlan auto-configure agent-circuit-identifier

- Use to create a VLAN subinterface that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets.
- Include this command in the base profile for a dynamic VLAN subinterface.
- Example

  host1(config-profile)#**vlan auto-configure agent-circuit-identifier**

- Use the **no** version to disable creation of VLAN subinterfaces based on agent-circuit-identifier information.

vlan description

- Use to assign a description to VLAN subinterfaces that are created with this profile.
- You can use a maximum of 64 characters for the description or to name the alias.

- Example

  host1(config-profile)#**vlan description test1**

- Use the **no** version to remove the VLAN description.

vlan policy

- Use to assign a VLAN policy list to an interface.
- For more information about keywords, see **vlan policy** on page 591.
- Example

  host1(config-profile)#**vlan policy input VlanPolicy33 statistics enabled preserve**

- Use the **no** version to remove the association between a policy list and an interface or a profile.

vlan profile

- Use to add a nested profile assignment to a base profile for a dynamic VLAN subinterface.
- A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the VLAN subinterface.
- Examples

  host1(config-profile)#**vlan profile pppoe vlanProfilePppoe**
  host1(config-profile)#**vlan profile ip vlanProfileIP**

- Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

vlan service-profile

- Use to specify a service profile name for a dynamic VLAN and to enter Service Profile Configuration mode. Service profiles contain user and password information, and are used in route maps for subscriber management and to authenticate subscribers with RADIUS.
- You can specify a service profile name with up to 80 alphanumeric characters.
- Example

  host1(config)#**vlan service-profile vlanClass1Service**
  host1(config-service-profile)#

- Use the **no** version to delete the service profile.

Assigning a Profile to an Interface

Use the **profile** command from Interface Configuration mode when you assign a profile to an interface.

For static PPP interfaces, you can assign only a profile for IP encapsulations. For static ATM 1483 subinterfaces, you can assign one profile for each bridged Ethernet, IP, PPP, and PPPoE encapsulation. For static VLAN subinterfaces, you can assign one profile for each IP or PPPoE encapsulation. You can also use the default keyword **any**, which applies to any autoconfigured encapsulation that does not have specific profile assignment.

For example, the following commands cause the router to use ProfileB when an IPoA packet is received, and to use ProfileA for any other received encapsulation that is autoconfigured. When you omit the keyword, it defaults to **any**.

```
host1(config-subif)#profile any ProfileA
host1(config-subif)#profile ip ProfileB
```

To assign a profile to an interface:

1. Configure a physical interface.

   ```
   host1(config-if)#interface atm 2/1.10
   ```

2. Configure a PVC by specifying the VCD, the VPI, the VCI, and the encapsulation type.

   ```
   host1(config-subif)#atm pvc 10 100 22 aal5snap
   host1(config-subif)#atm pvc 10 100 22 aal5autoconfig
   ```

3. Apply an existing profile.

   ```
   host1(config-subif)#profile ip holland
   ```

4. Assign subscriber identification.

   ```
   host1(config-subif)#subscriber ip user ispname domain abc.com
   password 3fds9jpt
   ```

5. Enable the dynamic encapsulation type.

   ```
   host1(config-subif)#auto-configure ip
   ```

atm pvc

- Use to configure a PVC on an ATM interface. Select one of the following encapsulation options:
  - **aal5autoconfig**—Enables the autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed).
  - **aal5snap**—Specifies a LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram.
  - **aal5mux ip**—Specifies a VC multiplexed circuit. This option is used for IP only.

- Example

  host1(config-subif)#**atm pvc 6 0 11 aal5autoconfig**

- Use the **no** version to remove the specified PVC.

auto-configure

- Use to configure an ATM subinterface to support a dynamic interface. Specifies one or more types of dynamic encapsulation that the ATM 1483 subinterface detects and accepts.
- For detailed information about how to use this command, see **auto-configure** on page 455.
- Example 1—Enables autodetection for the bridged Ethernet encapsulation type using the default lockout time range, 1–300 seconds

  host1(config-subif)#**auto-configure bridgedEthernet**

- Example 2—Enables autodetection for the bridged Ethernet encapsulation type using a nondefault lockout time range of 3600–21600 seconds (1–6 hours)

  host1(config-subif)#**auto-configure bridgedEthernet lockout-time 3600 21600**

- Example 3—Disables encapsulation type lockout for the IP encapsulation type

  host1(config-subif)#**auto-configure ip lockout-time none**

- Example 4—Either command reenables encapsulation type lockout for the IP encapsulation type using the default lockout time range

  host1(config-subif)#**auto-configure ip**
  host1(config-subif)#**no auto-configure ip lockout-time**

- Example 5—Permanently locks out the PPP encapsulation type until the **auto-configure ppp** command is issued

  host1(config-subif)#**no auto-configure ppp**

- Use the **no** version to terminate detection of the specified encapsulation type or, if the **lockout-time** keyword is specified, to restore the lockout time range to its default value, 1–300 seconds.

profile

- Use to assign a profile to a static ATM 1483 or static PPP interface. The profile configuration is used to dynamically configure an upper bridged Ethernet, IP, PPP, or PPPoE interface.
- The default encapsulation type, **any**, applies to any autoconfigured encapsulation that does not have a specific profile assignment.
- Example

  host1(config-subif)#**profile ip holland**

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

subscriber

- Use to configure a local subscriber on the router to support authentication and configuration from RADIUS for a dynamic IPoA or bridged Ethernet interface.
- For detailed information about how to use this command, see **subscriber** on page 476.
- Example

  host1(config-subif)#**subscriber ip user-prefix charlie domain myisp password-prefix lucy**

- Use the **no** version to remove the subscriber.

Profile Configuration Examples

The following examples show different ways to configure profiles.

- This example configures a new profile with IP characteristics only.

  host1(config)#**profile ProfileA**
  host1(config-profile)#**ip mtu 1024**
  host1(config-profile)#**exit**

- This example shows a new profile configured with both IP and PPP characteristics.

  host1(config)#**profile ProfileB**
  host1(config-profile)#**ip mtu 512**
  host1(config-profile)#**ppp authentication chap**
  host1(config-profile)#**ppp keepalive 120**
  host1(config-profile)#**exit**

- This example shows a new profile configured with IP, PPP, and PPPoE characteristics.

  host1(config)#**profile ProfileC**
  host1(config-profile)#**ip mtu 1400**
  host1(config-profile)#**ppp authentication chap**
  host1(config-profile)#**ppp keepalive 60**
  host1(config-profile)#**pppoe sessions 64**
  host1(config-profile)#**exit**

- This example uses the profiles created in the previous three examples. It shows distinct profiles for each encapsulation, where the configuration of dynamic layers varies according to which incoming encapsulation the ATM 1483 subinterface detects. Autodetection is enabled for the IP encapsulation type with the default lockout time range, 1–300 seconds.

  host1(config)#**interface atm 4/0.1**
  host1(config-subif)#**atm pvc 10 100 22 aal5autoconfig**
  host1(config-subif)#**profile ip ProfileA**
  host1(config-subif)#**profile ppp ProfileB**
  host1(config-subif)#**profile pppoe ProfileC**
  host1(config-subif)#**subscriber ip user atm1 domain isp1 password atm1pw**
  host1(config-subif)#**auto-configure ip**
  host1(config-subif)#**auto-configure ppp**

```
host1(config-subif)#auto-configure pppoe
host1(config-subif)#exit
```

- This example also uses the three new profiles configured in the first three examples. It shows one profile being used for all encapsulations. The configuration of dynamic layers is the same regardless of incoming encapsulations detected by ATM. Only relevant profile attributes are used for whichever dynamic interface layers are actually constructed.

```
host1(config)#interface atm 4/0.2
host1(config-subif)#atm pvc 200 0 200 aal5autoconfig
host1(config-subif)#profile any ProfileC
host1(config-subif)#subscriber ip user atm2 domain isp2 password atm2pw
host1(config-subif)#auto-configure ip
host1(config-subif)#auto-configure ppp
host1(config-subif)#auto-configure pppoe
host1(config-subif)#exit
```

- This example uses the three new profiles configured in the first three examples, and is implicitly assigned via the **any** encapsulation wildcard. Configuration of dynamic layers is the same regardless of incoming encapsulation detected by ATM. Autodetection is enabled for the IP encapsulation type with a lockout time range of 3600–7200 seconds (1–2 hours).

```
host1(config)#interface atm 4/0.3
host1(config-subif)#atm pvc 300 0 300 aal5autoconfig
host1(config-subif)#profile any ProfileC
host1(config-subif)#subscriber ip user atm2 domain isp3 password atm3pw
host1(config-subif)#auto-configure ip lockout-time 3600 7200
host1(config-subif)#auto-configure ppp
host1(config-subif)#auto-configure pppoe
host1(config-subif)#exit
```

- This example uses the profile configured in the first example. Autodetection is enabled for the bridged Ethernet encapsulation type with a lockout time range of 3600–21600 seconds (1–6 hours).

```
host1(config)#interface atm 4/0.3
host1(config-subif)#atm pvc 300 0 300 aal5autoconfig
host1(config-subif)#profile bridgedEthernet ProfileA
host1(config-subif)#subscriber bridgedEthernet user atm3 domain isp1
password fjdkei
host1(config-subif)#auto-configure bridgedEthernet lockout-time 3600 21600
```

Scripts and Macros

Scripts and macros are intended to reduce the management of static interfaces. Because dynamic interfaces have static lower layers, you can use scripts and macros to configure the static portion of all dynamic interfaces.

A script or macro can specify the static interface by using the **interface**, **auto-configure**, **subscriber**, or **profile** commands. These commands enable you to configure the interface as dynamic and to specify configuration sources for the dynamic upper layers. These files can then be executed by the router as though the commands were entered at the terminal.

- Scripts—You can create script files containing a series of CLI commands. The resulting script can be executed via the **configure file** command.
- Macros—You can create macros that generate and execute CLI commands. You first write macros on a computer and then copy them to the router. You issue the **macro** command from the CLI to execute both local macros or macros stored remotely. The **macro** command is available from all command modes. See *JUNOSe System Basics Configuration Guide, Chapter 10, Writing CLI Macros*.

NOTE: For a list of vendor-specific attributes (VSAs) that apply to dynamic interfaces, see *JUNOSe Broadband Access Configuration Guide, Chapter 1, Configuring Remote Access*.

Monitoring Upper-Layer Dynamic Interfaces and Profiles

You can use the **show** commands described in this section to monitor configurations created with dynamic interfaces and profiles.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

show atm aal5 interface

- Use to display information about a configured ATM AAL5 interface.
- Field descriptions
  - AAL5 Interface operational status—Operational status of the AAL5 interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the AAL5 interface operational status in hh:mm:ss format
  - SNMP trap link-status—Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
  - Auto configure ATM 1483 status—Whether the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface is enabled or disabled

- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of incoming errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- OutDiscards—Number of outgoing packets discarded on this interface

- Example

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46

SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

show atm subinterface

- Use to display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface.
- To specify an ATM subinterface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM subinterface for E120 and E320 routers, use the *slot*/*adapter*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647

- To display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface, use the **summary** keyword.
- To display status information only for ATM subinterfaces with a specific operating status, use the **status** keyword with one of the following status values. (See the Status field description for an explanation of these values.)
  - dormant
  - dormantLockout
  - down
  - lowerLayerDown
  - notPresent
  - up
- To display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values, use the **atm** *slot/port/vpi/vci* format (for ERX-7xx models, ERX-14xx models, and ERX-310 routers) or the *slot/adapter/port/vpi/vci* format (for E120 and E320 routers) to identify the ATM subinterface (Example 5).

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Field descriptions
  - Interface—Interface identifier
  - ATM-Prot—One of the following ATM protocol types:
    - RFC-1483—Multiprotocol encapsulation over AAL5
    - NBMA—Nonbroadcast multiaccess interface
    - ATM/MPLS—Local ATM passthrough interface
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual circuit (or channel) identifier
  - Circuit Type—Type of circuit: PVC
  - Encap—Administered encapsulation method based on what was configured with the **atm pvc** command
  - MTU—Maximum transmission unit size for the interface

- Status—One of the following ATM 1483 subinterface states:
  - absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled.
  - dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface.
  - dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout.
  - down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured.
  - lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down.
  - up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.
- Interface Type—Type of ATM 1483 subinterface: dynamic or static
- Auto configure status—Setting of the autoconfiguration feature
  - dynamic—Autodetection is on; the router automatically detects the next upper interface
  - static—Autodetection is off
- Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE
- Detected 1483 encapsulation—If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
- Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
- Interface types in lockout—Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none

- Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Next—Lockout time for the router to use for the next lockout event, in seconds
- Assigned profile—For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name
- Subscriber info—Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
- SNMP trap link-status—Trap link status: enabled or disabled
- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- InPacketsUnknownProtocol—Number of incoming packets with an unknown protocol type
- OutDiscards—Number of outgoing packets discarded on this interface

- Example 1—Displays the current state of all ATM subinterfaces

```
host1#show atm subinterface
                                    Circuit                              Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status            Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102   0 102 PVC     AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103   0 103 PVC     AUTO  9180 dormant        Static
3 interface(s) found
```

- Example 2—Displays summary information for all ATM subinterfaces shown in Example 1

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

- Example 3—Displays status information for all ATM subinterfaces in the dormantLockout state

```
host1#show atm subinterface status dormantLockout
                                    Circuit                              Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status            Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
1 interface(s) found
```

- Example 4—Displays the current state of a specific ATM subinterface

```
host1#show atm subinterface atm 2/0.101
                                    Circuit                              Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status            Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP

Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------- --- ---- ------- ------- ----
IP                                1   30      16       7   30
BridgedEnet                     900 3600       0       0  900
PPP                               1  300       0       0    1
PPPoE                             1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  1904
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         14
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

- Example 5—Displays the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values

```
host1#show atm subinterface atm 0/0/0/101
                                  Circuit                    Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                            1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled

InPackets:                5119
InBytes:                  358672
OutPackets:               5107
OutBytes:                 357510
InErrors:                 0
OutErrors:                0
InPacketDiscards:         3
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

show atm vc

- Use to display a summary of all configured ATM VCs and reserved VC ranges.
- You can specify one or more of the following keywords individually or in combination:
  - **vpi**—Displays VCs on a specific VPI
  - **category**—Displays VCs that have a specific service category
  - **status**—Displays VCs with a certain status
- To display only a summary of all reserved VC ranges on the router, specify the **reserved** keyword with no other keywords. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces.
- Field descriptions
  - Interface—Interface identifier
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - VCD—Virtual circuit descriptor

- Type—Type of circuit: PVC
- Encap—Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
- Category—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
- Rx/Tx Peak—Peak rate, in Kbps
- Rx/Tx Avg—Average rate, in Kbps
- Rx/Tx Burst—Maximum number of cells that can be burst at the peak cell rate
- Status—State of the virtual circuit: Up or Down
- Start VPI—Starting virtual path identifier (inclusive) of the reserved VC range
- Start VCI—Starting virtual circuit identifier (inclusive) of the reserved VC range
- End VPI—Ending virtual path identifier (inclusive) of the reserved VC range
- End VCI—Ending virtual circuit identifier (inclusive) of the reserved VC range

- Example 1—Displays all VCs and reserved VC ranges on the router

```
host1#show atm vc
                                        Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found


Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- Example 2—Displays all reserved VC ranges on the router

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

show columns

- Use to display static and dynamic interface counts for each interface column.
- Counts for PPP and PPPoE interface columns are updated when the PPP layer comes up.
- Counts for bridged Ethernet and IP over ATM columns are updated when the ATM layer comes up.
- Field descriptions
  - Type—Interface type
  - Total—Total number of interfaces on this column
  - Static—Number of static interfaces on this column
  - Dynamic—Number of dynamic interfaces on this column
- Example

```
host#show columns
                  Interface columns:
                  ------------------
        Type           Total      Static    Dynamic
-------------------- ---------- ---------- ----------
Bridged Ethernet              4          2          2
IP over ATM                   4          2          2
PPP                          22         12         10
PPPoE                        10          5          5
```

show pppoe interface

- Use to display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column.
- The following field descriptions and example include only the portion of the **show pppoe interface** command display relevant to lockout configuration for PPPoE clients. For more information about using this command, see **show pppoe interface** in *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*.
- Field descriptions
  - Lockout Configuration (seconds)—Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column
    - Min—Minimum lockout time, in seconds
    - Max—Maximum lockout time, in seconds
    - Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout
    - Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451

- Example

```
host1#show pppoe interface atm 3/0.101
. . .
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

show pppoe interface lockout-time

- Use to display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface.
- Field descriptions
  - PPPoE interface—Specifier for the PPPoE interface
  - Lockout Configuration (seconds)—Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column
    - Min—Minimum lockout time, in seconds
    - Max—Maximum lockout time, in seconds
    - Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout
    - Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see *Guidelines for Configuring Encapsulation Type Lockout* on page 451
  - Client Address—Source MAC address of the PPPoE client
  - Current—Current lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
  - Next—Lockout time that the router uses for the next lockout event, in seconds
- Example

```
host1#show pppoe interface atm 3/0.101 lockout-time
PPPoE interface ATM 3/0.101
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockout: 0
 Total clients in lockout grace period: 0
Client Address Current Elapsed Next
-------------- ------- ------- ----
0090.1a10.165e       0       0    5
```

show pppoe subinterface

- Use to display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session. You can then specify this MAC address in the **pppoe clear lockout interface** command to clear the lockout condition for the PPPoE client.
- To display configuration, status, and statistics information, including the source MAC address of the PPPoE client, use the **full** keyword.
- The following field descriptions and example include only the portion of the **show pppoe subinterface** command display relevant to the source MAC address for PPPoE clients. For more information about using this command, see **show pppoe subinterface** in *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*.
- Field descriptions
  - PPPoE subinterface—Specifier for the PPPoE subinterface
  - source MAC address—MAC address of the PPPoE client associated with the dynamic PPPoE subinterface column
- Example

```
host1#show pppoe subinterface full
...
    PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
...
```

show profile

- Use to display information about profiles.
- To display information about a specific profile, use the **name** keyword.
- To display a list of profiles configured on the router, use the **brief** keyword.
- Field descriptions
  - Profile—Name of the profile that is displayed
  - 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—Name of the virtual router (VR) assigned to the profile; interfaces created by the profile are attached to this VR
  - Directed Broadcast—Enabled or disabled
  - ICMP Redirects—Enabled or disabled
  - Access Route Addition—Enabled or disabled
  - Network Address Translation—Enabled or disabled; domain location (inside or outside)
  - Source-Address Validation—Enabled or disabled
  - Ignore DF Bit—Enabled or disabled
  - Filter Option Packets—Router filters out packets with IP options; enabled or disabled

- Administrative MTU—MTU size configured on the profile
- TCP MSS value—Maximum segment size for TCP SYN packets traveling through the interface
- Inactivity Timer—Inactivity timer setting; enabled or disabled
- Route Map Name—Route map applied to the IP interface subscriber; enabled or disabled
- 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
- IGMP—Enabled or disabled
- static-groups—Displays address of any static groups configured for IGMP
- Input policy—Name of input policy and whether statistics are enabled or disabled
- Output policy—Name of output policy and whether statistics are enabled or disabled
- PPP Keepalive—PPP keepalive period, in seconds
- PPP Magic Number—Enabled or disabled
- PPP Magic Number Mismatch—Indicates whether the router is configured to ignore the LCP peer magic number and retain the PPP connection when the peer has not negotiated an LCP magic number: ignore (ignore the peer magic number mismatch and retain the PPP connection), or reject (router terminates the PPP connection if it detects a peer magic number mismatch)
- PPP Peer DNS Priority—Enabled or disabled
- PPP Peer WINS Priority—Enabled or disabled
- PPP Authentication—Type of authentication configured: PAP, CHAP, or none
- PPP Authentication Router—Name of authentication virtual router
- PPP Negotiate MRU—MRU configured for the profile
- PPP Packet Log—Enabled or disabled
- PPP State Log—Enabled or disabled
- PPP Chap Challenge Length—Minimum and maximum Chap Challenge length
- PPP Passive Mode—Enabled or disabled
- PPP Multilink—Enabled or disabled
- PPP IPCP netmask option—Enabled or disabled
- PPP AAA Profile—AAA profile associated with this PPP interface
- PPP Multilink Fragmentation—Enabled or disabled
- PPP Multilink Fragment Size—Multilink fragment size for this PPP interface
- PPP Multilink Reassembly—Enabled or disabled

- PPP Multilink Mrru—Multilink MRRU value for this PPP interface
- PPP Initiate IP—Initiation of IPv4 over this PPP interface; enabled or disabled
- PPP Initiate IPv6—Initiation of IPv6 over this PPP interface; enabled or disabled
- PPPoE Max Sessions—Maximum number of PPPoE subinterfaces that can be on an interface
- PPPoE Always-offer—Router offers to set up a session for the client, even if the router has insufficient resources to establish a session; enabled or disabled
- PPPoE Remote-Circuit-Id—The router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM); enabled or disabled
- PPPoE Log PPPoeControlPacket—Enabled or disabled
- PPPOE duplicate-protect—Enabled or disabled
- PPPoE ACNAME—Access concentrator name
- PPPoE URL—URL sent in PADM message to PPPoE clients
- PPPoE MOTM—Message of the minute sent in the PADM message to PPPoE clients
- PPPoE Service-Name Table—Name of the PPPoE service name table, if configured for the specified profile
- ATM1483 Auto-configure—Whether autodetection of the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) is enabled or disabled for a dynamic ATM 1483 subinterface
- ATM1483 lockout (seconds)—Encapsulation type lockout setting for the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) configured on a dynamic ATM 1483 subinterface
  - range—Minimum lockout time–maximum lockout time, in seconds
  - no lockout—Encapsulation type lockout is disabled
- ATM1483 PVC circuit type—Encapsulation setting for the PVC configured on a dynamic ATM 1483 subinterface
  - aal5autoconfig—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
  - aal5mux ip—VC-based multiplexed circuit for IP only
  - aal5snap—LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram
- ATM1483 PVC service category—Service type setting for the PVC configured on a dynamic ATM 1483 subinterface: UBR (the default), UBR PCR, NRT-VBR, RT-VBR, or CBR
- ATM1483 PVC Peak rate—Peak cell rate (PCR), in Kbps, for the PVC configured on a dynamic ATM 1483 subinterfaces
- ATM1483 PVC Avg rate—Average rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as sustained cell rate (SCR)

- ATM1483 PVC Burst size—Length in cells of the burst for the PVC configured on a a dynamic ATM 1483 subinterface; also referred to as maximum burst size (MBS)
- ATM1483 Description—Text description assigned to ATM 1483 subinterfaces that are created with this profile
- ATM1483 Advisory Rx Speed—Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E-series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
- ATM1483 PVC OAM Administrative status—Status of OAM F5 loopback cell generation (for VC integrity) on a circuit created with this profile: enabled or disabled
- ATM1483 PVC OAM Loopback frequency—Number of seconds between transmissions of OAM F5 end-to-end loopback cells on a circuit created with this profile
- ATM1483 Ip Subscriber information—Subscriber login information for the specified dynamic interface type
- ATM1483 Profile—Name of the profile assigned to the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE); these profiles are referenced in the base profile for a dynamic ATM 1483 subinterface as nested profile assignments
- VLAN Auto-configure—Whether auto detection of the specified upper-interface encapsulation type (IP or PPPoE) is enabled or disabled for a dynamic VLAN subinterface
- VLAN Advisory Rx Speed—Configured advisory receive speed, in Kbps, for the dynamic VLAN subinterface; the E-series LAC sends this value to the LNS in the RX Connect-Speed AVP [38]
- VLAN Advisory Tx Speed—Configured advisory speed, in Kbps, for the dynamic VLAN subinterface.
- VLAN Description—Text description assigned to VLAN subinterfaces that are created with this profile
- VLAN Profile—Name of the profile assigned to the specified upper-interface encapsulation type (IP or PPPoE); these profiles are referenced in the base profile for a dynamic VLAN subinterface as nested profile assignments
- VLAN Service Profile—Service profile name for a VLAN
- VLAN Svlan Ethertype—Ethertype that the packet must use this to create the dynamic VLAN subinterface
- Bridged Ethernet Mtu—MTU size configured for a dynamic bridged Ethernet interface
- Bridged Ethernet Service Profile—Name of the IP service profile associated with the interface profile for this dynamic bridged Ethernet interface
- IPv6 Unnumbered interface—Name of interface without a specific address
- IPv6 Router—Router name or default
- IPv6 Src-Addr Validation—Source-Address Validation; enabled or disabled
- IPv6 Administrative MTU—MTU size
- IPv6 ND Enabled—State of the Neighbor Discovery; enabled or disabled

- IPv6 ND ManagedConfig—State of the Neighbor Discovery router advertisement managed flag; enabled or disabled
- IPv6 ND OtherConfig—State of the Neighbor Discovery router advertisement other config flag; enabled or disabled
- IPv6 ND SuppressRa—Status IPv6 router advertisement suppression; enabled or disabled
- IPv6 ND RaInterval—Interval (in seconds) of the Neighbor Discovery router advertisement
- IPv6 ND RaLifeTime—Lifetime (in seconds) of the Neighbor Discovery router advertisement
- IPv6 ND ReachableTime—Amount of time (in milliseconds) that the neighbor is expected to remain reachable
- IPv6 ND RaPrefix—Configured prefixes for Neighbor Discovery router advertisement
- IPv6 ND ValidLifetime—Amount of time (in seconds) that the router advertises the IPv6 prefix as valid
- IPv6 ND PreferredLifetime—Amount of time (in seconds) that the router advertises the specified IPv6 prefix as preferred
- IPv6 ND PrefixOnLink—State of the on-link flag; enabled or disabled
- IPv6 ND PrefixAutoConfig—State of the use the specified prefix for IPv6 autoconfiguration; enabled or disabled

- Example 1—Displays configuration information for a profile assigned to a dynamic interface

```
host1#show profile name pppoeProfile
Profile                     : pppoeProfile
Unnumbered interface on     : loopback 1
Router                      : default
Directed Broadcast          : Disabled
ICMP Redirects              : Disabled
Access Route Addition       : Enabled
Network Address Translation: Disabled
Source-Address Validation   : Disabled
Ignore DF Bit               : Disabled
Filter Option Packets       : Disabled
Administrative MTU          : 1500
TCP MSS value               : 0
Inactivity Timer            : Disabled
Route Map Name              : Disabled
Auto Detect                 : Disabled
Auto Configure              : Disabled

IGMP                        : Enabled
  static-groups             :
  Input policy:  bobb statistics enabled
  Output policy: bobb statistics enabled

PPP Keepalive                : 30
PPP Magic Number             : enabled
PPP Magic Number Mismatch    : ignore
PPP Peer DNS Priority        : disabled
PPP Peer WINS Priority       : disabled
PPP Authentication           : pap/chap
PPP Authentication Router    :
```

```
PPP Negotiate MRU           : (use lower layer MRU)
PPP Packet Log              : disabled
PPP State Log               : disabled
PPP Chap Challenge Length   : 16 - 32
PPP Passive Mode            : disabled
PPP Multilink               : disabled
PPP IPCP Netmask Option     : disabled
PPP AAA Profile             :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly    : disabled
PPP Multilink Mrru          : (use MRU)
PPP Initiate IP             : disabled
PPP Initiate IPv6           : disabled
PPPoE Max Sessions          : 2
PPPoE Always-offer          : Disabled
PPPoE Remote-Circuit-Id     : Enabled
PPPoE Log PPPoeControlPacket: Disabled
PPPoE duplicate-protect     : Enabled
PPPoE ACNAME                : CYM9876
PPPoE URL                   : http://www.urlofinterest.com
PPPoE MOTM                  : goodmorning
PPPoE Service-Name table    : myServiceTable1
```

- Example 2—Displays configuration information for a base profile assigned to a dynamic ATM 1483 subinterface

```
host1#show profile name atm1483BaseProfile
ATM1483 Auto-configure ip                 : disabled
ATM1483 Auto-configure bridgedEthernet    : disabled
ATM1483 Auto-configure ppp                : enabled
ATM1483 lockout (seconds) ppp             : range : 1-300
ATM1483 Auto-configure pppoe              : enabled
ATM1483 lockout (seconds) pppoe           : range : 1-300
ATM1483 PVC circuit type         :  aal5autoconfig
ATM1483 PVC service category     :  Nrt-Vbr
ATM1483 PVC Peak rate : 10000, Avg rate : 2000, Burst size : 500
ATM1483 Description              :  VC_atm1483
ATM1483 Advisory Rx Speed        :  2000000000

ATM1483 PVC OAM Administrative status: enabled
ATM1483 PVC OAM Loopback frequency:    30

ATM1483 Ip Subscriber information:
  user             : elaine
  domain           : jpeterman.com
  password         : putty
ATM1483 IP Profile               : none assigned
ATM1483 Bridged Ethernet Profile : none assigned
ATM1483 PPP Profile              : none assigned
ATM1483 PPPoE Profile            : pppoeProfile
```

- Example 3—Displays configuration information for a base profile assigned to a dynamic VLAN subinterface

```
host1#show profile name vlanProfile
VLAN Auto-configure ip             : enabled
VLAN Auto-configure pppoe          : enabled
VLAN Svlan Ethertype               : auto-configure
VLAN Advisory Rx Speed             : 100 Kbps
VLAN Advisory Tx Speed             : 2500 Kbps
VLAN Description                   : testing
VLAN IP Profile                    : ipProfile
```

```
VLAN PPPoE Profile                : pppoeProfile
VLAN Service Profile              : none assigned
Bridged Ethernet Mtu              : 1971
Bridged Ethernet Service Profile  : eastServiceProfile
```

- Example 4—Displays profile configuration information related to IPv6 Neighbor Discovery router advertisement

```
host1#show profile name ipv6Profile
IPv6 Unnumbered interface  : loopback 0
IPv6 Router                : default
IPv6 Src-Addr Validation   : Disabled
IPv6 Administrative MTU    : 0
IPv6 ND Enabled            : Enabled
IPv6 ND ManagedConfig      : Disabled
IPv6 ND OtherConfig        : Enabled
IPv6 ND SuppressRa         : Disabled
IPv6 ND RaInterval         : 50
IPv6 ND RaLifeTime         : 1800
IPv6 ND ReachableTime      : 0
IPv6 ND RaPrefix           : 1234::/64
IPv6 ND ValidLifetime      : 60
IPv6 ND PreferredLifetime  : 60
IPv6 ND PrefixOnLink       : Enabled
IPv6 ND PrefixAutoConfig   : Enabled
```

show vlan subinterface

- Use to display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router.
- Use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router.
- Use the **vlan** or **svlan** keywords to display information about specific VLAN IDs or S-VLAN IDs.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface
  - Status—Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
  - MTU—Maximum allowable size (in bytes) of the MTU for the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID value for the VLAN subinterface
  - Ethertype—S-VLAN Ethertype value, if configured
  - Type—Type of VLAN subinterface
    - Static—VLAN or S-VLAN subinterface was configured statically
    - Dynamic—VLAN or S-VLAN subinterface was configured dynamically
  - Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: IP or PPPoE

- Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: IP, PPPoE, or (if no packet has been received) none
- Interface types in lockout—Encapsulation types currently experiencing lockout: IP, PPPoE, or none
- Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Next—Lockout time for the router to use for the next lockout event, in seconds
- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received on the VLAN or S-VLAN subinterface
  - Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface
  - Packets—Number of packets sent on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses

        - Errors—Total number of errors in all ARP packets
        - Discards—Total number of discarded ARP packets
    - Total VLAN interfaces—Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; only this field appears when you specify the **summary** keyword

- Example 1—Displays full status and configuration information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
ATM 3/0.1.2              Up         1522 ----     11       ----      Static
ATM 3/0.1.3              Up         1522 ----     12       ----      Static
ATM 3/1.1.1              Up         1522 ----     13       ----      Static
ATM 3/1.1.2              Up         1522 ----     14       ----      Static
ATM 3/2.1.1              Down       1526 4        255      0x9100    Static
FastEthernet 4/5.1       Up         1522 ----     1        ----      Dynamic
6 vlan subinterfaces found
```

- Example 2—Displays full status and configuration information for the specified VLAN subinterface

```
host1#show vlan subinterface fastEthernet 4/5.1
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 4/5.1       Up         1522 ----     1        ----      Dynamic
1 vlan subinterface found
```

- Example 3—Displays full status and configuration information for the specified S-VLAN ID

```
host1#show vlan subinterface svlan id 100 53
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 0/0.1       Up         1526 100      53       0x9100    Static
FastEthernet 4/6.1       Up         1526 100      53       0x9100    Dynamic

2 vlan subinterfaces found
```

- Example 4—Displays full status and configuration information for the specified dynamic VLAN subinterface

```
host1#show vlan subinterface fastEthernet 4/6.1000053
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 4/6.1000053  Up         1526 100      53       0x9100    Dynamic

Auto configure interface(s)   : IP  PPPoE
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
IP                               1 300       0       0    1
PPPoE                            1 300       0       0    1
```

```
  In: Bytes 1040, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
  Out: Bytes 984, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

Troubleshooting PPP and PPPoE Dynamic Interfaces

You can issue the **profile-reassign** command to help you use PPP and PPPoE packet–logging capabilities to debug and troubleshoot PPP and PPPoE dynamic interfaces. To use the **profile-reassign** command, you must access Privileged Exec mode at privilege level 5 or higher.

The **profile-reassign** command enables you to reassign the profile currently assigned to the specified encapsulation type for the specified ATM 1483 subinterface. In effect, you swap the currently assigned nondebug profile for a debug profile that has identical attributes with the addition of one or more PPP or PPPoE packet–logging attributes enabled.

To troubleshoot PPP and PPPoE dynamic interfaces:

1. Create a debug profile that includes the same attributes as an existing nondebug profile, with the addition of one or more PPP or PPPoE packet–logging attributes enabled.

   Observe the following guidelines when you create the debug profile:

   - Because PPP and PPPoE packet logging is performed at log severity 7 (debug priority), configure a destination such as the console to log severity level 7 and issue the **log here** command to enable packet capture using the debug profile you created.

   - Before you reassign the debug profile to the ATM 1483 subinterface, make sure that the number of PPP dynamic interfaces has not already exceeded the maximum number of aggregate dynamic and static PPP interfaces for which you can log PPP packets. For more information about this and other system maximums, see *JUNOSe Release Notes, Appendix A, System Maximums*.

   For details about creating and using profiles, see *Configuring a Dynamic Interface from a Profile* on page 483.

2. Access Privileged Exec mode at privilege level 5 or higher.

   ```
   host1>enable 5
   Password: *********
   host1#
   ```

NOTE: The router prompts you for a password only if you have configured a password to control access to Privileged Exec mode. For details about setting passwords to access various command privilege levels, see *JUNOSe System Basics Configuration Guide, Chapter 9, Passwords and Security*.

3. From Privileged Exec mode, issue the **profile-reassign** command to replace the nondebug profile currently assigned to the specified encapsulation type for the specified ATM 1483 interface with the debug profile created in Step 1.

   You must specify one of the following encapsulation types to which the debug profile applies: **ppp**, **pppoe**, or **any**. You can use the **any** encapsulation type if neither the **ppp** encapsulation type nor the **pppoe** encapsulation type has an existing profile assignment. For example:

   host1#**profile-reassign atm 2/0.101 ppp pppLogConfig**

NOTE: Issuing the **profile-reassign** command causes the router to tear down any dynamic interfaces that exist above the ATM 1483 subinterface. After the profile is reassigned, the router restores the interfaces based on the necessary client reconnections. If the ATM 1483 subinterface is currently shut down, issuing the **profile-reassign** command does not reestablish the interface connection.

4. (Optional) Use the appropriate **show** command to verify the profile reassignment. For example:

   host1#**show atm subinterface atm 2/0.101**

   When you reassign a debug profile to an ATM 1483 subinterface, the reassignment is stored in NVS and persists after a reboot. If you issue the **show atm subinterface** or **show configuration** command after the profile is reassigned, these commands display the new profile assignment.

5. (Optional) To restore the initial (nondebug) profile assignment after you troubleshoot the dynamic interface, issue the **profile-reassign** command again using the name of the nondebug profile. For example:

   host1#**profile-reassign atm 2/0.101 ppp pppConfig**

enable

- Use to move from User Exec to Privileged Exec mode.
- In Privileged Exec mode, you can access all other user interface modes. From here you can configure, monitor, and manage all aspects of the router.
- Optionally, you can specify one of the following privilege levels; the default level is 10.
  - **0**—The user can execute the **help**, **enable**, **disable**, and **exit** commands.
  - **1**—The user can execute commands in User Exec mode plus commands at level 0.
  - **5**—The user can execute Privileged Exec **show** commands plus the commands at levels 1 and 0.

  - **10**—The user can execute all commands except support commands, which may be provided by Juniper Networks Customer Service.
  - **15**—The user can execute support commands.
- Set a password for this mode by using either the **enable password** or the **enable secret** command in Global Configuration mode. Doing so protects the router from any unauthorized use.
- After a password is set, anyone trying to use Privileged Exec mode is prompted to provide the password.
- Example

  ```
  host1>enable 5
  Password:*******
  host1#
  ```

- There is no **no** version.

profile-reassign

- Use to reassign the profile currently assigned to the specified encapsulation type for the specified ATM 1483 interface. For troubleshooting purposes, use the **profile-reassign** command to swap the currently assigned profile for one that has PPP or PPPoE packet–logging attributes enabled.
- This command is available from Privileged Exec mode at privilege level 5 or higher.
- Specify one of the following keywords:
  - **ppp**—Specifies a PPP encapsulation type to which the profile applies
  - **pppoe**—Specifies a PPPoE encapsulation type to which the profile applies
  - **any**—Specifies any autoconfigured encapsulation that does not have a specific profile assignment; valid only if neither the **ppp** encapsulation type nor the **pppoe** encapsulation type has an existing profile assignment
- Specify a profile name of up to 80 alphanumeric characters.
- Example 1—Facilitates debugging for the **ppp** encapsulation type by swapping profile pppConfig for profile pppLogConfig, which includes PPP packet–logging attributes

  ```
  host1#profile-reassign atm 2/0.101 ppp pppLogConfig
  WARNING: Execution of this command will cause all dynamic interfaces over
  atm 2/0.101 to be torn-down.
  Proceed with profile reassignment? [confirm] yes
  Profile pppConfig replaced by profile pppLogConfig for ppp.
  ```

- Example 2—Facilitates debugging for the **any** encapsulation type by swapping profile anyConfig for profile anyLogConfig, which includes both PPP and PPPoE packet–logging attributes

  ```
  host1#profile-reassign atm 3/0.101 any anyLogConfig
  WARNING: Execution of this command will cause all dynamic interfaces over
  atm 3/0.101 to be torn-down.
  Proceed with profile reassignment? [confirm] yes
  Profile anyConfig replaced by profile anyLogConfig for any.
  ```

- Example 3—Restores the initial (nondebug) profile assignment for the **ppp** encapsulation type shown in Example 1. Assuming that PPP packet logging is not configured in profile pppConfig, this command also disables logging for the interface

  ```
  host1#profile-reassign atm 2/0.101 ppp pppConfig
  WARNING: Execution of this command will cause all dynamic interfaces over
  atm 2/0.101 to be torn-down.
  Proceed with profile reassignment? [confirm] yes
  Profile pppLogConfig replaced by profile pppConfig for ppp.
  ```

- There is no **no** version.

Chapter 16

Configuring Dynamic Interfaces Using Bulk Configuration

This chapter explains dynamic interfaces and describes the procedures for configuring them on E-series routers.

This chapter contains the following sections:

Overview

Before you begin configuring dynamic interfaces in bulk, review the concepts described in this section.

Like upper-layer dynamic interfaces, bulk-configured dynamic interfaces are created automatically and transparently through the receipt of data over a lower-layer link, such as an ATM virtual circuit (VC) or a virtual LAN (VLAN) using autodetection. The layers of a dynamic interface are created based on the packets received on the link and can be configured through any one of the following:

- RADIUS authentication (through PPP or ATM 1483)
- Profiles
- A combination of RADIUS authentication and profiles

You create and configure each layer of a static interface manually through an existing configuration mechanism such as the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

For more information about dynamic interfaces, autodetection, and RADIUS, see *Overview* in *Chapter 15, Configuring Dynamic Interfaces*.

Bulk Dynamic Interface Configurations

E-series routers support dynamic interfaces on two types of static interfaces: ATM and VLAN. This chapter provides configuration information for ATM and then for VLANs.

E-series routers support dynamic ATM 1483 subinterfaces over static ATM interfaces.

E-series routers support the following types of dynamic interfaces over VLAN major interfaces:

- Dynamic VLAN subinterface over static VLAN major interface
- IP over dynamic VLAN subinterface
- IP over PPPoE over dynamic VLAN subinterface

Internet Protocol version 4 (IPv4) is supported for all bulk-configured dynamic interface columns over dynamic ATM 1483 subinterfaces and over dynamic VLAN subinterfaces.

Currently, Internet Protocol version 6 (IPv6) is supported only when PPP or MLPPP is the layer immediately below the IPv6 layer in the interface column. IPv6 is *not* supported directly over dynamic ATM 1483, dynamic bridged Ethernet, or dynamic VLANs. Bulk-configured dynamic interface columns that support IPv6 include the following:

- Dynamic IPv6 over dynamic PPP over dynamic ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over dynamic ATM 1483
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over dynamic ATM 1483
- Dynamic IPv6 over dynamic PPP over dynamic PPPoE over dynamic VLAN
- Dynamic IPv6 over dynamic MLPPP over dynamic PPPoE over dynamic VLAN

For more information about IPv4, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*. For more information about IPv6, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 2, Configuring IPv6*.

Profiles

You can use profiles to configure dynamic interfaces over ATM and VLAN interfaces. A *profile* is a set of characteristics that can be dynamically assigned to interfaces. By using a profile, you reduce the management of a large number of interfaces by applying a set of characteristics to multiple interfaces.

When you are configuring a large number of interfaces with the same attributes at the higher layers, you can use a profile to factor out all the common attributes of each layer into one place. This action affects one or more dynamic layers of the interface column. After you define the static lower layers, you assign a profile to the highest static layer of the interface column.

When a dynamic interface is configured, the configuration data received from the RADIUS authentication server typically overrides configuration data obtained from a profile.

The **atm atm1483 auto-configure** command specifies the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic ATM 1483 subinterface. For flexibility, the router provides the ability to configure an ATM 1483 subinterface with distinct profile assignments for each encapsulation type supported by the **atm atm1483 auto-configure** command. For more information about using this command, see **atm atm1483 auto-configure** on page 552.

In contrast to dynamic ATM 1483 subinterfaces, dynamic VLAN subinterfaces support recognition and creation of simultaneous IP and PPPoE upper dynamic interface types. The **vlan auto-configure** command identifies the encapsulation type. For flexibility, the router provides the ability to configure a VLAN subinterface with distinct profile assignments for each encapsulation type supported by the **vlan auto-configure** command. For more information about using this command, see **vlan auto-configure** on page 589.

For more information about configuring profiles, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

ATM Oversubscription for Bulk-Configured VC Ranges

You can take advantage of oversubscription of bulk-configured ATM VCs. The router supports oversubscription of bulk-configured VC ranges when you create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.

Oversubscription of bulk-configured VC ranges works in a similar, but not identical, manner to oversubscription of static ATM 1483 subinterfaces that support dynamic upper-layer encapsulation types. For more information, see *ATM Oversubscription for Dynamic Interfaces* in *Chapter 15, Configuring Dynamic Interfaces*.

Bulk-Configured VC Ranges

An active bulk-configured VC range is associated with a dynamic ATM 1483 subinterface that supports a dynamic upper-layer encapsulation type. For ATM line modules that support VC oversubscription, the maximum number of active bulk-configured VCs per line module is less than the maximum number of individual VCs created from the total number of bulk-configured VC ranges that the line module supports. For information about configuring dynamic ATM 1483 subinterfaces with bulk-configured VC ranges, see *Configuring ATM 1483 Dynamic Subinterfaces* on page 541.

When the maximum number of active bulk-configured VCs has been reached, the router prevents all additional subscribers associated with the remaining inactive bulk-configured VCs from connecting to the line module until one of the following conditions occurs:

- At least one currently active subscriber logs out, which causes the router to tear down the dynamic interface column for that subscriber. Although the dynamic ATM 1483 subinterface and its associated VC remain configured on the router, the subinterface becomes inactive and can be replaced by one of the bulk-configured VCs waiting to become active.

- The router tears down at least one dynamic interface column in its entirety, which involves administratively shutting down the associated dynamic ATM 1483 subinterface.

When either of these conditions occurs, the router enables the first inactive bulk-configured VC that receives traffic to connect to the router as a replacement for the subscriber that logged out.

Example

Consider an ATM line module that supports a maximum of 32,000 individual VCs created from bulk-configured VC ranges, of which only 8000 VCs can be active at any one time. If all 32,000 bulk-configured VCs attempt to connect to the router, only the first 8000 VCs to receive traffic are able to log in, generate dynamic subinterface columns, and become active. When a subscriber connected through one of these active VCs logs out, the router enables the first of the remaining 24,000 inactive bulk-configured VCs that receives traffic to connect. The router replaces the inactive dynamic ATM 1483 subinterface and associated VC that remain after the subscriber logout with a new dynamic ATM 1483 subinterface and its newly activated circuit.

Combination of Static ATM 1483 Subinterfaces and Bulk-Configured VC Ranges

ATM line modules are sometimes configured with a combination of static ATM 1483 subinterfaces and bulk-configured VC ranges. In these configurations, both the static ATM 1483 subinterfaces and bulk-configured VC ranges can support active subinterfaces. The combined total of active static ATM 1483 subinterfaces, and active dynamic ATM 1483 subinterfaces created from bulk-configured VC ranges, cannot exceed the maximum number of active subinterfaces supported by the line module.

The number of active dynamic subinterfaces created from the bulk-configured VC ranges is limited by both of the following:

- The number of static ATM subinterfaces that exist on the line module, which cannot exceed the maximum number of configured ATM 1483 subinterfaces supported on the line module.
- The number of static ATM subinterfaces that are active on the line module, which cannot exceed the maximum number of active ATM 1483 subinterfaces supported on the line module.

Example

Consider an ATM line module that supports a maximum of 8000 active ATM 1483 subinterfaces. The module has 4000 static ATM 1483 subinterfaces configured, all of which are active, and 8000 individual VCs created from bulk-configured VC ranges. Because the 4000 static ATM 1483 subinterfaces are already active, the router enables only 4000 of the bulk-configured VCs to create dynamic ATM 1483 subinterface columns and become active, yielding a combined total of 8000 active subinterfaces on the line module. The router prevents the remaining 4000 inactive bulk-configured VCs from connecting and becoming active until at least one subscriber connected through an active ATM subinterface logs out, thereby making the associated subinterface inactive and eligible for replacement.

Platform Considerations

You can configure dynamic interfaces on the following E-series routers:

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

Module Requirements

For information about the modules that support dynamic interfaces on ERX-14xx models, ERX-7xx 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 dynamic interfaces.

For information about the modules that support dynamic interfaces 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 dynamic interfaces.

Interface Specifiers

The configuration task examples in this chapter use the *slot*/*port*[.*subinterface*] format to specify the physical interface that you want to configure to support dynamic interfaces. However, the interface specifier format that you use depends on the router that you are using.

For ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX-7xx model, ERX-14xx model, or ERX-310 router.

```
host1(config)#interface atm 0/1.10
```

For E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

For more information about supported interface types and specifiers on E-series routers, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.

References

For more information about RADIUS, consult the following resources:

- DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006)
- RFC 2865—Remote Authentication Dial In User Service (RADIUS) (June 2000)
- RFC 2866—RADIUS Accounting (June 2000)
- RFC 3046—DHCP Relay Agent Information Option (January 2001)

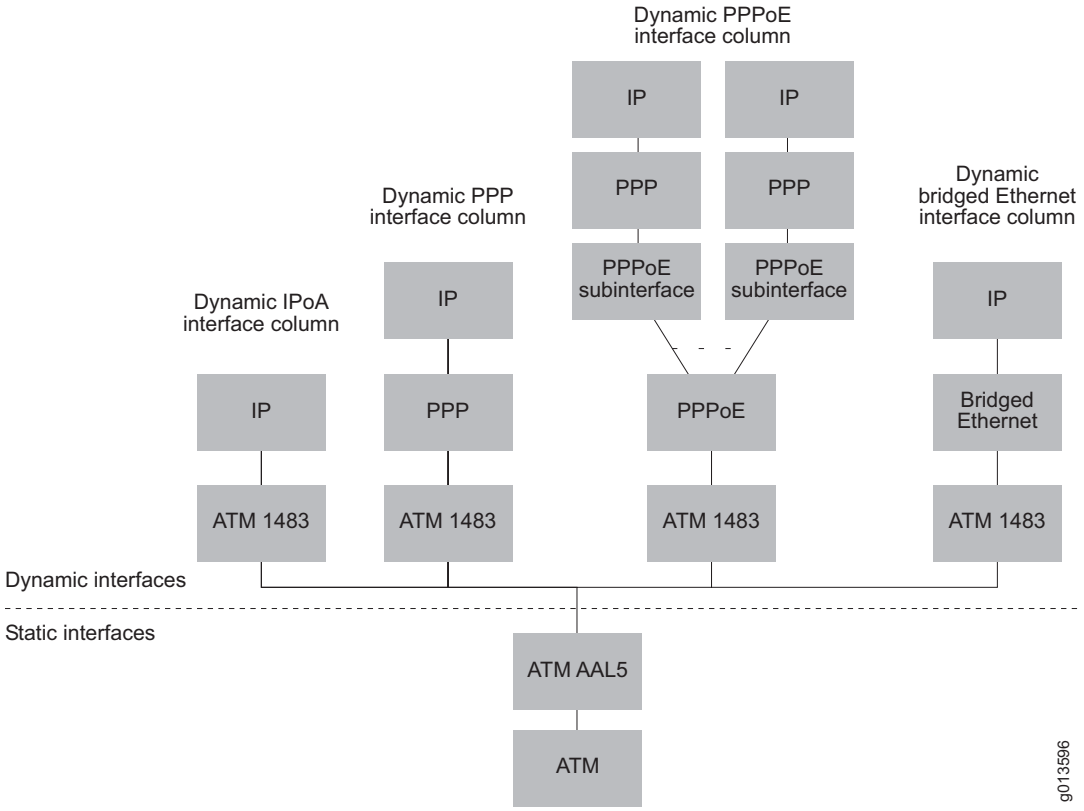
Configuring ATM 1483 Dynamic Subinterfaces

E-series routers support configuration of dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces over ATM. The dynamic ATM 1483 subinterface can perform autodetection and dynamic creation of the following upper-layer encapsulation types:

- Bridged Ethernet
- IP
- PPP
- PPPoE

Figure 50 shows the dynamic upper-interface columns supported by dynamic ATM 1483 subinterfaces, and indicates which layers in the columns are static and dynamic.

Figure 50: Dynamic Interface Columns over Dynamic ATM 1483 Subinterfaces



About Configuring Dynamic ATM 1483 Subinterfaces

This section introduces important concepts that you need to understand before you configure dynamic ATM 1483 subinterfaces.

Overview and Benefits

When you use dynamic interfaces over static ATM 1483 subinterfaces, you must configure the ATM interface and each ATM 1483 subinterface, including the ATM PVC and the attributes of the subinterface. Subinterface attributes include profile assignments, autoconfiguration settings, and subscriber configurations.

By contrast, when you use dynamic ATM 1483 subinterfaces over static ATM AAL5 interfaces, you use a process called *bulk configuration* to configure a range of ATM PVCs that support dynamic interfaces. On receipt of an incoming packet on the virtual circuit, the router dynamically creates the ATM 1483 subinterface. As part of the configuration process, you create an ATM 1483 base profile, which can optionally include nested profile assignments, to define the attributes required to configure the dynamic ATM 1483 subinterface and the dynamic upper-layer encapsulation types built over it.

Bulk configuration provides an efficient and timesaving way to specify a range of ATM PVCs for dynamic ATM 1483 subinterfaces. Because bulk configuration requires significantly less configuration of the router, it results in reduced output when you issue the **show configuration** command to display the current router configuration.

Dynamic ATM 1483 subinterfaces function identically to static ATM 1483 subinterfaces, except for the manner in which they are created and configured. The creation of dynamic upper-layer encapsulation types is essentially the same regardless of whether they are configured over static ATM 1483 subinterfaces or dynamic ATM 1483 subinterfaces.

ATM 1483 Base Profiles

To configure a dynamic ATM 1483 subinterface over a static ATM AAL5 interface, you must create a base profile. The base profile includes one or more of the following attributes for the ATM 1483 subinterface, listed alphabetically:

- advisory-rx-speed—Sets an advisory receive speed for ATM 1483 subinterfaces that are created with this base profile. For information, see **atm atm1483 advisory-rx-speed** on page 552.
- atm pvc—Applies encapsulation, traffic-shaping, and OAM parameters to the range of ATM PVCs configured on the ATM AAL5 interface for use by the dynamic ATM 1483 subinterface. For information, see **atm pvc** on page 556.
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic ATM 1483 subinterface. For information, see **atm atm1483 auto-configure** on page 552.
- atm class-vc—Specifies the VC class assigned to the bulk-configured VC ranges created on the dynamic ATM 1483 subinterfaces associated with this base profile. For information, see **atm class-vc** on page 555.

- description—Assigns a description to ATM 1483 subinterfaces that are created with this base profile. For information, see **atm atm1483 description** on page 553**.** You can then set up the router to send this description to AAA by using the **atm atm1483 export-subinterface-description** command, as described in *Sending Interface Descriptions to AAA* in *Chapter 1, Configuring ATM*.

- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the ATM 1483 subinterface. For information, see **atm atm1483 profile** on page 553.

- subscriber—Configures a local subscriber for a dynamic upper-interface encapsulation type. For information, see **atm atm1483 subscriber** on page 553.

You can override the base profile assignment for a single ATM PVC that exists within a bulk-configured VC subrange with a profile that includes debugging attributes. This feature is useful for troubleshooting problems with the ATM 1483 dynamic subinterface columns created on the specified PVC. For more information, see *Overriding Base Profile Assignments* on page 546.

Nested Profile Assignments

The configuration for each dynamic upper-interface encapsulation type might differ, depending on the column type built by the router. To manage these differences, you can include one or more nested profile assignments within the ATM 1483 base profile. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface encapsulation type. You can create different profiles for each upper-interface encapsulation type, or you can create a single profile that includes attributes for multiple encapsulation types.

For example, the following commands create a base profile named atm1483BaseProfile with two nested profile assignments. The first nested profile assignment references an IP profile named atm1483ProfileIp, and the second nested profile assignment references a PPP profile named atm1483ProfilePpp.

```
host1(config)#profile atm1483BaseProfile
host1(config-profile)#atm atm1483 profile ip atm1483ProfileIp
host1(config-profile)#atm atm1483 profile ppp atm1483ProfilePpp
```

In this example, atm1483ProfileIp and atm1483ProfilePpp have different IP configurations depending on the dynamic interface column constructed. For an IP over ATM (IPoA) dynamic interface column, the router uses the IP attributes in atm1483ProfileIp. For an IP over PPP over ATM dynamic interface column, the router uses the IP attributes in atm1483ProfilePpp.

The concepts that apply to profiles created for upper-interface encapsulation types configured over static ATM 1483 subinterfaces also apply to profiles created for upper-interface encapsulation configured over dynamic ATM 1483 subinterfaces. For information about creating profiles for upper-interface encapsulation types, see *Chapter 15, Configuring Dynamic Interfaces*.

Additional Profile Characteristics for Upper Interfaces

In addition to ATM 1483 attributes and nested profile assignments, the base profile for a dynamic ATM 1483 subinterface can also include individual characteristics for several upper-interface encapsulation types, provided that no nested profile assignment for the specified encapsulation type is in the base profile. If, on the other hand, a nested profile assignment for this encapsulation type exists in the base profile, the router obtains all characteristics for that encapsulation type from the nested profile and not from the base profile.

For lists of the characteristics for each supported upper-interface encapsulation type, see *Profile Characteristics* in *Chapter 15, Configuring Dynamic Interfaces*.

Bulk Configuration of VC Ranges

When you create a static ATM 1483 subinterface, you must configure a permanent virtual circuit (PVC), also known as a virtual circuit (VC). The ATM protocol requires one or more VCs over which data traffic is transmitted to higher layers in the protocol stack.

Similarly, dynamic creation of ATM 1483 subinterfaces requires you to configure a range of ATM PVCs on the ATM AAL5 interface and assign a name to this range. Each VC range consists of one or more nonoverlapping VC subranges. A VC subrange is a group of VCs that resides within the virtual path identifier (VPI) and virtual circuit identifier (VCI) ranges you specify.

The process of configuring a VC range for a dynamic ATM 1483 subinterface is referred to as *bulk configuration*. You create a bulk configuration by issuing the **atm bulk-config** command. For example, the following commands create an ATM 1483 bulk configuration named myBulkConfig on the specified ATM AAL5 interface.

```
host1(config)#interface atm 2/0
host1(config-if)#atm bulk-config myBulkConfig vc-range 0 3 101 1100
vc-range 4 7 201 700
```

In this example, the **atm bulk-config** command configures a VC range made up of two VC subranges. The first subrange, with VPIs 0–3 and VCIs 101–1100, configures 1000 VCs on each of four VPIs, for a total of 4000 VCs. The second subrange, with VPIs 4–7 and VCIs 201–700, configures 500 VCs on each of four VPIs, for a total of 2000 VCs. The entire myBulkConfig VC range configures a combined total of 6000 VCs.

NOTE: For information about the maximum number of ATM 1483 bulk configurations supported per router, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

After you issue the **atm bulk-config** command, the router provisions all circuits in the specified VC range at the same time. This provisioning can take several seconds, depending on the number of VCs being created. The router does not dynamically create the ATM 1483 subinterface for the circuit until it receives incoming data traffic on the circuit.

After you create a named VC range, you cannot remove the underlying ATM AAL5 interface until you issue the **no atm bulk-config** command to remove the VC range from that interface.

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see *Configuring F4 OAM* in *Chapter 1, Configuring ATM*.

Bulk Configuration and VC Classes

You can assign a previously configured VC class to a bulk-configured VC range. A *VC class* is a set of attributes for virtual circuits that can include the service category, encapsulation method, F5 OAM options, and Inverse ARP. Using VC classes to configure VC attributes provides the following benefits:

- VC classes enable you to classify and group VCs based on the OAM and traffic requirements of their associated subscribers.
- When subscriber requirements change, a VC class is easier and less time-consuming to modify than individual PVC attributes.

To assign a VC class to a bulk-configured VC range, you use the **atm class-vc** command from Profile Configuration mode to associate the VC class to a base profile. Issuing this command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this base profile.

For details about configuring and using VC classes, including information about how precedence levels affect how the router determines attributes values for dynamically created circuits, see *Configuring ATM VC Classes* in *Chapter 1, Configuring ATM*. For information about how to use the **atm vc-class** command to assign a VC class to a base profile, see **atm class-vc** on page 555.

NOTE: Using the **atm class-vc** command inside a nested profile that is referenced in a base profile has no effect on the bulk-configured VC ranges associated with the base profile. The router accepts only those VC class assignments that are configured in a base profile and ignores any VC class assignments made in a nested profile.

Bulk Configuration and CAC

You cannot create a bulk-configured VC range on an ATM interface on which you have configured connection admission control (CAC). Conversely, you cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range.

If you are upgrading to the current JUNOSe software release from a lower-numbered release, configurations that use CAC and bulk configuration on the same ATM interface continue to work. However, we recommend that you disable CAC on these ATM interfaces to ensure continued compatibility with future JUNOSe releases.

For information about how to use the **atm cac** command to configure CAC, see *Setting Optional Parameters* in *Chapter 1, Configuring ATM*.

Dynamic Interface Creation

After you configure the ATM 1483 base profile and create the range of VCs on the ATM AAL5 interface, you associate these two components by assigning the base profile to the VC range with the **profile atm1483 bulk-config-name** command.

As a final step, you must issue the **auto-configure atm1483** command. This command configures the ATM AAL5 interface to support autodetection of the ATM 1483 dynamic encapsulation type.

When the router receives an incoming data packet on a circuit, it dynamically creates the ATM 1483 subinterface, using the attributes specified in the base profile. After examining the contents of the data packet, the router dynamically creates the required interface columns above the ATM 1483 subinterface, using the configuration attributes contained in the nested profiles, if specified, or in the base profile itself.

Overriding Base Profile Assignments

You can use the **profile atm1483 bulk-config-name pvc** command to assign an overriding profile to a single ATM PVC that exists within a bulk-configured VC subrange. The VC subrange that encompasses the PVC must have been previously configured with the **atm bulk-config** command for use by a dynamic ATM 1483 subinterface. After you assign the overriding profile, the router uses the information in this profile instead of the information in the previously assigned base profile to create any subsequent ATM 1483 dynamic subinterface columns on the specified PVC.

Overriding the base profile assignment for an ATM PVC with a profile that includes debugging attributes enables you to troubleshoot problems with ATM 1483 dynamic subinterface columns created on the specified PVC. The overriding profile, like the original base profile, can include ATM 1483 attributes, nested profile assignments, and individual characteristics for dynamic upper-interface encapsulation types.

For configuration instructions and examples, see *Configuring Overriding Profile Assignments* on page 558.

NOTE: See *JUNOSe Release Notes*, *Appendix A*, *System Maximums* for information about the maximum number of overriding profile assignments currently supported per router.

Changing VC Subranges

You can add, remove, modify, merge, disable, and enable VC subranges within an existing bulk-configured VC range. Previously, changes to VC subranges were possible only if you removed the VC range and then configured it again with different subrange values. The ability to make changes to VC subranges without first having to remove the entire VC range avoids potentially disrupting all subscribers on existing dynamic ATM 1483 subinterfaces associated with the deleted VC range.

For configuration instructions and examples, see *Changing VC Subranges* on page 563.

Static ATM Interfaces Within VC Subranges

You can configure a static ATM interface with an ATM PVC whose VPI and VCI addresses fall within an existing bulk-configured VC subrange. Conversely, you can also create a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface. Previously, configurations that caused VPI/VCI address conflicts between a static ATM interface and a bulk-configured VC subrange were prohibited on the router.

In certain ATM network configurations, you might need to transparently forward traffic from selected circuits with unrelated addresses to another location in the network. The ability to create a static ATM interface on a circuit within a bulk-configured VPI/VCI address range is particularly useful when you use ATM layer 2 services over MPLS with Martini encapsulation to forward the traffic from the selected circuits. You must create the interface stack for ATM layer 2 statically and define the configuration parameters individually on a per-interface basis.

The following rules apply when you configure either a static ATM interface within an existing bulk-configured VC subrange, or a subrange that includes an existing static ATM interface:

- All of the following ATM configurations are supported on the static ATM interface: ATM layer 2 services over MPLS including local cross-connects, point-to-point connections, and nonbroadcast multiaccess (NBMA) connections.
- Static ATM interfaces and circuits defined within a bulk-configured VC subrange are stored in NVS and preserved after a reboot.
- The base profile associated with the VC subrange does not apply to any statically defined ATM interfaces that fall within the subrange.
- If a VC subrange includes a statically defined ATM interface, overriding profile assignments configured for the same VPI/VCI address as a statically defined ATM interface become inactive until the static ATM 1483 subinterface is removed. The overriding profile becomes active again when you remove the static ATM 1483 subinterface. You can display the current operational status (active or inactive) of overriding profile assignments by using the **show atm bulk-config** command.
- Operations that add, remove, modify, merge, disable, or enable VC subranges within a bulk-configured VC range do not affect any static ATM interfaces defined within the VC subrange.

- You cannot create a static ATM circuit if the VPI/VCI address conflicts with an existing ATM 1483 dynamic subinterface column. Such a configuration would disrupt subscribers already connected to the router via the dynamic subinterface.

- You cannot create a static ATM interface with a VPI/VCI address that falls within a range of circuits reserved for use by the MPLS downstream-on-demand label distribution method.

- You cannot configure CAC on a static ATM interface within an existing bulk-configured VC subrange. Conversely, you cannot create a bulk-configured VC subrange that includes a static ATM interface on which CAC is configured. (For information about how to use the **atm cac** command to configure CAC, see *Setting Optional Parameters* in *Chapter 1, Configuring ATM*.)

For configuration information and examples, see *Configuring Static ATM Interfaces Within VC Subranges* on page 568.

Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations

In configurations of dynamic IP over dynamic PPP over a dynamic (bulk-configured) ATM 1483 subinterface, the router sends an LCP terminate request packet to a PPPoA CPE device in response to receipt of an IPv4-over-PPP data packet or an IPv6-over-PPP data packet when the dynamic ATM 1483 subinterface transitions to a dormant state due to an ungraceful subscriber logout. This action terminates stale PPPoA subscribers and causes the CPE to restart LCP negotiations. This behavior is always in effect on the router and does not require CLI or SNMP configuration.

The implementation of this feature for dynamic ATM 1483 subinterfaces is almost identical to the implementation for static ATM 1483 subinterfaces, with the following difference:

- For *static* ATM 1483 subinterfaces, the restart of LCP negotiations by the CPE causes the router to re-create the dynamic PPP and IP upper-layer interfaces above the static ATM 1483 subinterface.

- For *dynamic* ATM 1483 subinterfaces, the receipt of a PPP data packet from the CPE causes the router to re-create only the dynamic ATM 1483 subinterface to send the LCP terminate request packet, but not the dynamic PPP and IP upper-layer interfaces above the dynamic ATM 1483 subinterface. The router re-creates the dynamic PPP and IP upper-layer interfaces when the CPE restarts LCP negotiations.

For details about the operation and benefits of this feature, see *Terminating Stale PPPoA Subscribers and Restarting LCP Negotiations* in *Chapter 15, Configuring Dynamic Interfaces*, which describes the router behavior for static ATM 1483 subinterfaces.

Authenticating Subscribers on Dynamic Bridged Ethernet over Dynamic ATM Interfaces

You can use either of the following methods to configure and manage RADIUS authentication for IP subscribers on dynamic bridged Ethernet over dynamic ATM 1483 subinterfaces:

- The **atm atm1483 subscriber** command
- The subscriber management application

The **atm atm1483 subscriber** command *does not support* running stateful SRP switchover (high availability) on the router. Therefore, the configuration method you choose depends on whether stateful SRP switchover is or is not running on your router.

Configuration Method Using atm atm1483 subscriber Command

When you use the **atm atm1483 subscriber** command, as described in **atm atm1483 subscriber** on page 553, to configure IP subscribers on dynamic bridged Ethernet over dynamic ATM 1483 subinterface columns to support RADIUS authentication, the **atm atm1483 subscriber** command provides the subscriber's authentication parameters. The dynamic ATM 1483 subinterface acts as the authenticating layer that establishes a session with RADIUS and passes the subscriber's locally configured username and password information to the RADIUS server.

However, if your router is running stateful SRP switchover (high availability), the use of the **atm atm1483 subscriber** command in this configuration might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. To bypass this limitation, you can use the subscriber management application to configure IP subscribers on dynamic bridged Ethernet interfaces.

Configuration Method Using Subscriber Management Application

You can use the JUNOSe subscriber management application to configure and manage IP subscribers associated with a dynamic bridged Ethernet interface column. The subscriber management application uses an IP service profile to manage and authenticate IP subscribers with RADIUS. An IP service profile contains user and password information, and is used in a route map for subscriber management and to authenticate subscribers with RADIUS.

In this configuration, the IP service profile provides the subscriber's authentication parameters, and the subscriber management application acts as the authenticating layer to obtain information from RADIUS for configuration of dynamic IP subscribers. To assign the IP service profile to the interface profile from which the dynamic bridged Ethernet interface is created, you use the **bridge1483 service-profile** command in Profile Configuration mode.

If stateful SRP switchover is disabled or not running on your router, you can continue to use the **atm atm1483 subscriber** command to configure IP subscribers on dynamic bridged Ethernet interfaces to support RADIUS authentication.

Alternatively, you can use the subscriber management application to create and configure dynamic IP interfaces regardless of whether stateful SRP switchover is running on the router. In addition, using subscriber management enables you to take advantage of several useful features such as the IP inactivity timer.

In the event that an interface profile for a dynamic bridged Ethernet interface includes the **atm atm1483 subscriber** command to configure a local subscriber as well as the **bridge1483 service-profile** command to reference an IP service profile, the values specified with the **atm atm1483 subscriber** command take precedence. The router ignores the values in the IP service profile in this case.

For details about using the subscriber management application to configure RADIUS authentication for IP subscribers on dynamic bridged Ethernet interfaces, see *Authenticating Subscribers on Dynamic Bridged Ethernet over Static ATM Interfaces* and *Configuring Subscriber Management for IP Subscribers on Dynamic Bridged Ethernet Interfaces* in *Chapter 15, Configuring Dynamic Interfaces*. The information in these sections, which explains how to use subscriber management to achieve the same functionality as the **subscriber** command without adversely affecting stateful SRP switchover, applies equally to the **atm atm1483 subscriber** command.

For more information about using the subscriber management application, see *JUNOSe Broadband Access Configuration Guide, Chapter 23, Configuring Subscriber Management*.

Configuring a Dynamic ATM 1483 Subinterface

To configure a dynamic ATM 1483 subinterface:

1. (Optional) Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic ATM 1483 subinterface.

   These profiles are referenced in the base profile for the dynamic ATM subinterface as nested profile assignments. For detailed instructions on creating profiles, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

2. Create the base profile for the dynamic ATM 1483 subinterface by assigning the profile a name.

   host1(config)#**profile atm1483BaseProfile**

   This command accesses Profile Configuration mode, which enables you to configure attributes in the base profile.

3. Define attributes for the ATM 1483 subinterface in the base profile.

   a. Apply traffic-shaping parameters to the VC range on the ATM AAL5 interface.

   b. Configure the ATM 1483 subinterface for autodetection of the PPP upper-interface encapsulation type.

c. Configure the ATM 1483 subinterface for autodetection of the IP upper-interface encapsulation type using a nondefault lockout time range of 3600–7200 seconds (1–2 hours).

d. Configure a subscriber for the IP upper-interface encapsulation type.

e. Configure a description for ATM 1483 subinterfaces that are created with this base profile.

f. Set an advisory speed for ATM subinterfaces that are created with this base profile.

g. Assign a VC class to the bulk-configured VC ranges created on the dynamic ATM 1483 subinterfaces associated with this base profile. You must issue the **exit** command from Profile Configuration mode for the VC class association to take effect.

```
host1(config-profile)#atm pvc aal5autoconfig cbr 10000
host1(config-profile)#atm atm1483 auto-configure ppp
host1(config-profile)#atm atm1483 auto-configure ip lockout-time 3600 7200
host1(config-profile)#atm atm1483 subscriber ip user-prefix joesmith
domain myisp password-prefix abc123
host1(config-profile)#atm atm1483 description VC_atm1
host1(config-profile)#atm atm1483 advisory-rx-speed 2000
host1(config-profile)#atm class-vc premium-subscriber-class
host1(config-profile)#exit
```

4. (Optional) In the base profile, create nested profile assignments for the upper-interface encapsulation types, and include additional profile characteristics for other encapsulation types as needed.

   For example, the following commands configure nested profile assignments for the PPP and IP upper-interface encapsulation types, and define additional attributes for the PPPoE upper-interface encapsulation type.

```
host1(config-profile)#atm atm1483 profile ppp myPppProfile
host1(config-profile)#atm atm1483 profile ip myIpProfile
host1(config-profile)#pppoe duplicate-protection
host1(config-profile)#pppoe sessions 3000
```

5. Exit Profile Configuration mode.

6. Configure the ATM and ATM AAL5 interface.

```
host1(config)#interface atm 5/0
```

7. Configure a range of VCs on the static ATM AAL5 interface, and assign a name to this range. This operation can take several minutes to complete, depending on the number of VCs being configured.

NOTE: For information about the maximum number of ATM 1483 bulk configurations supported per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

For example, the following command creates a VC range named myBulkConfig made up of two VC subranges that configure a total of 5,000 virtual circuits.

host1(config-if)#**atm bulk-config myBulkConfig vc-range 0 2 101 1100 vc-range 3 6 201 700**

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see *Configuring F4 OAM* in *Chapter 1, Configuring ATM*.

8. Assign the base profile configured for the ATM 1483 subinterface to the VC range configured on the ATM AAL5 interface.

   host1(config-if)#**profile atm1483 bulk-config-name myBulkConfig atm1483BaseProfile**

9. Configure the ATM AAL5 interface to support autodetection of the ATM 1483 dynamic encapsulation type.

   host1(config-if)#**auto-configure atm1483**

atm atm1483 advisory-rx-speed

- Use to set an advisory receive speed for ATM 1483 subinterfaces that are created with the profile that you are configuring. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *JUNOSe Broadband Access Configuration Guide, Chapter 12, Configuring an L2TP LAC* for additional information about the advisory receive speed.
- The range is 0–2147483647 kbps.
- Example

  host1(config-profile)#**atm atm1483 advisory-rx-speed 2000**

- Use the **no** version to restore the default behavior—the RX speed is not sent to the LNS.

atm atm1483 auto-configure

- Use to specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic ATM 1483 subinterface.
- Include this command in the base profile for a dynamic ATM 1483 subinterface.
- For the bridged Ethernet, IP, PPP, and PPPoE encapsulation types, you can optionally specify the lockout time range for the encapsulation type. For more information, see *Encapsulation Type Lockout* on page 449.

- Examples

  host1(config-profile)#**atm atm1483 auto-configure ip lockout-time 3600 7200**
  host1(config-profile)#**atm atm1483 auto-configure pppoe**

- Use the **no** version to terminate detection of the specified encapsulation type.

atm atm1483 description

- Use to assign a text description for ATM 1483 subinterfaces that are created with the profile that you are configuring.
- The description can be up to 255 characters.
- Example

  host1(config-profile)#**atm atm1483 description VC_atm1**

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

atm atm1483 profile

- Use to add a nested profile assignment to a base profile for a dynamic ATM 1483 subinterface.
- A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the ATM 1483 subinterface.
- Example

  host1(config-profile)#**atm atm1483 profile pppoe atm1483ProfilePppoe**

- Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

atm atm1483 subscriber

- Use to configure a local subscriber for a dynamic upper-interface encapsulation type configured over a dynamic ATM 1483 subinterface. A subscriber supports authentication and configuration from the RADIUS server.
- Optionally, you can include this command in the base profile for a dynamic ATM 1483 subinterface.
- When you configure a subscriber, you must specify the following:
  - *upperInterfaceType*—Type of dynamic interface, **bridgedEthernet** or **ip**
  - *userNameUsage*—How the dynamic interface uses the username for authentication purposes
    - **user**—Use the name as specified.
    - **user-prefix**—Use the name as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the username string. The username format is *userName.slot.port.vpi.vci*. The resulting username string is then used to authenticate the subscriber with the RADIUS server.
  - *userName*—RADIUS username
  - *domainName*—Domain name

- You can optionally supply password information:
  - *passwordUsage*—How the dynamic interface uses the password for authentication purposes
    - **password**—Use the password as specified.
    - **password-prefix**—Use the password as a prefix to the interface physical location. The router automatically postpends the physical location of the user to the password string. The password format is *password.slot.port.vpi.vci*. The resulting password string is then used to authenticate the subscriber with the RADIUS server.
  - *password*—RADIUS password
- If your router is running stateful SRP switchover (high availability), the use of the **atm atm1483 subscriber** command to configure RADIUS authentication for subscribers on dynamic bridged Ethernet interfaces might suspend stateful SRP switchover on the router or prevent stateful SRP switchover from becoming active. For more information about using the subscriber management application to bypass this limitation, see *Authenticating Subscribers on Dynamic Bridged Ethernet over Dynamic ATM Interfaces* on page 549.
- Example 1

  host1(config-profile)#**atm atm1483 subscriber ip user-prefix boston01 domain myisp password-prefix abc123**

- Example 2

  host1(config-subif)#**atm atm1483 subscriber bridgedEthernet user westford003 domain acmecorp.east password xyz123**

- Use the **no** version to remove the subscriber.

atm bulk-config

- Use to create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.

NOTE: For information about the maximum number of ATM 1483 bulk configurations supported per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

- Each VC range consists of one or more nonoverlapping VC subranges. A VC subrange is a group of VCs that resides within the VPI and VCI ranges you specify.
- You can configure multiple VC ranges on an ATM AAL5 interface.

NOTE: For information about the maximum number of VCs (sum of the VPI/VCI addresses within all VC subranges) that you can configure with the **atm bulk-config** command per line module and per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

NOTE: Do not use any reserved VCI values when configuring VCs with the **atm bulk-config** command. For information about reserved VCIs, see *Configuring F4 OAM* in *Chapter 1, Configuring ATM*.

- When you create a bulk-configured VC range, you must specify the following:
  - A name of up to 80 alphanumeric characters; this is also referred to as the bulk configuration name
  - The starting and ending VPI values (inclusive) for each VC subrange
  - The starting and ending VCI values (inclusive) for each VC subrange
- You can create a placeholder VC range by issuing the **atm bulk-config** command without specifying any subranges. You can assign a profile to this placeholder and add subranges to it later.
- You can add and remove individual VC subranges.
- You cannot remove a VC subrange if any dynamic ATM 1483 subinterfaces currently exist for any circuit within the subrange. Use the **atm bulk-config shutdown** command to remove dynamic ATM 1483 interfaces created within a subrange.
- Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.
- You can create a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface.
- You cannot create a bulk-configured VC range on an ATM interface on which you have configured CAC. Conversely, you cannot configure CAC on an ATM interface on which you have created a bulk-configured VC range. For information about configuring CAC, see *Setting Optional Parameters* in *Chapter 1, Configuring ATM*.
- Example 1—Configures a VC range named myBulkConfig with a single VC subrange containing VPIs 0–2 and VCIs 101–1100; this command configures a total of 3000 VCs

  host1(config-if)#**atm bulk-config myBulkConfig vc-range 0 2 101 1100**

- Example 2—Configures a VC range named myMultiBulkConfig with two VC subranges containing VPIs 0–1 and VCIs 101–600 (first subrange) and VPIs 3–5 and VCIs 201–3200 (second subrange); this command configures a total of 10,000 VCs

  host1(config-if)#**atm bulk-config myMultiBulkConfig vc-range 0 1 101 600 vc-range 3 5 201 3200**

- Use the **no** version to remove the specified VC range from the ATM AAL5 interface, to remove the specified subranges from the specified VC range, or to remove all subranges from the specified VC range. The **no** version also removes any overriding profile assignments for ATM PVCs within the deleted VC range or VC subrange.

atm class-vc

- Use to assign a previously configured VC class to a base profile for a dynamic ATM 1483 subinterface.
- Issuing this command applies the set of attributes in the specified VC class to all bulk-configured VC ranges that are dynamically created from this base profile.
- You must issue the **exit** command from Profile Configuration mode for the VC class association to take effect.

- Changes to a VC class specified in a base profile apply only to those PVCs that are dynamically created *after* the change is made. These changes do not apply to dynamic PVCs that were created prior to the VC class modification.
- Example

  host1(config-profile)#**atm class-vc gold-subscriber-class**
  host1(config-profile)#**exit**

- Use the **no** version to remove the VC class association with the base profile.

atm pvc

- Use to apply encapsulation, traffic-shaping, and OAM parameters to the range of ATM PVCs configured on an ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.
- Include this command in the base profile for a dynamic ATM 1483 subinterface.
- You must specify one of the following encapsulation types:
  - **aal5autoconfig**—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
  - **aal5snap**—Specifies a logical link control (LLC) encapsulated circuit; the LLC/Subnetwork Access Protocol (LLC/SNAP) header precedes the protocol datagram
  - **aal5mux ip**—Specifies a VC-based multiplexed circuit used for IP only
- You can optionally set the *peak*, *average*, and *burst* sizes. To use VBR-RT or VBR-NRT as the service type, you must specify each of these options.
- The default service type is UBR. To set a different service type, specify one of the following keywords:
  - **rt**—Selects VBR-RT as the service type; you can select **rt** only if you set the *peak*, *average*, and *burst* parameters
  - **cbr**—Selects CBR as the service type; you must set the CBR rate in Kbps
- You can optionally include the **oam** keyword and a number of seconds in the range 1–600 to enable generation of OAM F5 loopback cells on this circuit. This option enables VC integrity features that affect the operational state of the ATM PVC.
- Example

  host1(config-profile)#**atm pvc aal5autoconfig cbr 10000 oam 120**

- Use the **no** version to restore the default service type, UBR, on the VC range.

auto-configure atm1483

- Use to configure the static ATM AAL5 interface to support autodetection of an ATM 1483 dynamic interface type.
- You must issue this command to enable creation of a dynamic ATM 1483 subinterface.

- Example

  host1(config-if)#**auto-configure atm1483**

- Use the **no** version to terminate autodetection of the ATM 1483 encapsulation type.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- To specify an ATM interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.[*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 5/0/0.1**

- Use the **no** version to remove the interface or subinterface.

profile

- Use to create a base profile to configure attributes for a dynamic ATM 1483 subinterface.
- Specify a profile name of up to 80 alphanumeric characters.
- Example

  host1(config)#**profile atm1483BaseProfile**

- Use the **no** version to delete the specified profile if it is not being used by any existing VC subranges.

NOTE: If VC ranges are configured for the dynamic ATM 1483 subinterface associated with the base profile you want to delete, you must use the **no atm bulk-config** command to remove the VC ranges before you can use the **no profile** command to remove the associated base profile.

profile atm1483 bulk-config-name

- Use to assign the base profile configured for a dynamic ATM 1483 subinterface to the VC range configured on a static ATM AAL5 interface.
- You must specify both of the following:
  - Name assigned to the VC range on an ATM AAL5 interface, as specified in the **atm bulk-config** command
  - Name assigned to the base profile for a dynamic ATM 1483 subinterface
- Example

  host1(config-if)#**profile atm1483 bulk-config-name myBulkConfig atm1483BaseProfile**

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

Configuring Overriding Profile Assignments

Configuring overriding profile assignments includes the following tasks:

- Assigning an overriding profile to an ATM PVC within a bulk-configured VC subrange
- Removing an overriding profile assignment from an ATM PVC
- Removing overriding profile assignments from a VC range or VC subrange

The following sections describe how to perform these tasks.

Assigning an Overriding Profile to an ATM PVC

You can assign an overriding profile to a single ATM PVC within a bulk-configured VC subrange. Typically, the overriding profile includes debugging attributes to help you identify and troubleshoot problems with the ATM 1483 dynamic subinterface column created on the specified PVC.

To assign an overriding profile to an ATM PVC within a bulk-configured VC subrange:

1. Configure both of the following:

   - Base profile for the bulk-configured VC range on the static ATM AAL5 interface. The VC range consists of one or more VC subranges.

   - Overriding profile for an ATM PVC within a bulk-configured VC subrange

   For information about configuring profiles, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

2. Create a bulk-configured range of VCs on a static ATM AAL5 interface. The following commands create a bulk-configured VC range named myBulkConfig that consists of two VC subranges. The first subrange encompasses VPIs 0–4 and VCIs 21–1000. The second subrange encompasses VPIs 5–7 and VCIs 21–2000.

   host1(config)#**interface atm 4/0**
   host1(config-if)#**atm bulk-config myBulkConfig vc-range 0 4 21 1000 vc-range 5 7 21 2000**

3. Assign the previously configured base profile (atm1483BaseProfile) to the bulk-configured VC range.

   host1(config-if)#**profile atm1483 bulk-config-name myBulkConfig atm1483BaseProfile**

4. Assign the previously configured overriding profile to a single ATM PVC within the bulk-configured VC subrange. The following command assigns the overriding profile myDebugProfile to the PVC with VPI 0 and VCI 101. This PVC exists within the first VC subrange (VPIs 0–4 and VCIs 21–1000) configured in Step 2.

   host1(config-if)#**profile atm1483 bulk-config-name myBulkConfig pvc 0 101 myDebugProfile**

   The router now uses the information in the overriding profile instead of the information in the base profile to create subsequent ATM 1483 dynamic subinterface columns over this PVC.

5. (Optional) You can assign the same overriding profile to a different ATM PVC within the same VC subrange or within a different VC subrange. For example, the following command assigns the overriding profile myDebugProfile to the PVC with VPI 6 and VCI 901. This PVC exists within the second VC subrange (VPIs 5–7 and VCIs 21–2000) configured in Step 2.

   host1(config-if)#**profile atm1483 bulk-config-name myBulkConfig pvc 6 901 myDebugProfile**

NOTE: You can reverse the order of Step 3 and Step 4 with identical results. That is, you can assign the overriding profile to the ATM PVC and then assign the base profile to the entire VC range. In either case, you must first create the bulk-configured VC range with the **atm bulk-config** command.

6. Configure the ATM AAL5 interface to enable all bulk configurations and to support autodetection of the ATM 1483 dynamic encapsulation type.

   host1(config-if)#**auto-configure atm1483**

7. (Optional) Use the **show atm bulk-config** command to verify the overriding profile configuration.

   For more information about using this command, see **show atm bulk-config** on page 602.

Removing an Overriding Profile Assignment from an ATM PVC

After you troubleshoot the ATM 1483 dynamic subinterface column created on the specified PVC, make sure that you remove the overriding profile assignment to restore the original base profile assignment. This action ensures that subsequent ATM 1483 dynamic subinterface columns are created using the same attributes defined in the base profile.

To remove an overriding profile assignment from an ATM PVC within a bulk-configured VC range:

1. Remove the overriding profile assignment from the specified ATM PVC.

   host1(config-if)#**no profile atm1483 bulk-config-name myBulkConfig pvc 0 101**

2. Select the dynamic ATM 1483 subinterface on which the ATM 1483 dynamic subinterface column resides.

   host1(config)#**interface atm 4/0.101**

3. Use the **shutdown** command to disable the dynamic ATM 1483 subinterface. The **shutdown** command deletes the ATM 1483 dynamic subinterface column and removes the dynamic ATM 1483 subinterface.

   host1(config-subif)#**shutdown**

4. Send traffic over the specified PVC (VPI 0 and VCI 101) on the ATM AAL5 interface. This action re-creates the ATM 1483 dynamic subinterface column with the original base profile association.

   The router now uses the information in the base profile instead of the information in the overriding profile to create subsequent ATM 1483 dynamic subinterface columns for the specified PVC.

5. (Optional) Use the **show atm bulk-config** command to verify the removal of the overriding profile assignment.

   For more information about using this command, see **show atm bulk-config** on page 602.

Removing Overriding Profile Assignments from a VC Range or VC Subrange

When you issue the **no atm bulk-config** command to remove an entire VC range (and all VC subranges within that VC range), the router also removes any overriding profile assignments configured for PVCs within those VC subranges. For example, the following command removes the bulk-configured VC range named myBulkConfig and any overriding profile assignments for PVCs within the VC subranges belonging to myBulkConfig.

```
host1(config-if)#no atm bulk-config myBulkConfig
```

When you issue the **no atm bulk-config** command to remove a particular VC subrange in a bulk-configured VC range, the router also removes any overriding profile assignments for PVCs within that VC subrange. However, overriding profile assignments for PVCs within other VC subranges in the VC range remain intact. For example, the following command removes one VC subrange (VPIs 0–4 and VCIs 21–1000) and only those overriding profile assignments associated with this subrange.

```
host1(config-if)#no atm bulk-config myBulkConfig vc-range 0 4 21 1000
```

atm bulk-config

- Use to create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.
- For detailed information about how to use this command, see **atm bulk-config** on page 554.
- Example

  ```
  host1(config)#atm bulk-config test1 vc-range 0 1 101 600
  vc-range 3 5 201 3200
  ```

- Use the **no** version to remove the specified VC range from the ATM AAL5 interface, to remove the specified subranges from the specified VC range, or to remove all subranges from the specified VC range. The **no** version also removes any overriding profile assignments for ATM PVCs within the deleted VC range or VC subrange.

auto-configure atm1483

- Use to configure the static ATM AAL5 interface to enable all bulk configurations and support autodetection of the ATM 1483 dynamic encapsulation type.
- You must issue this command to enable creation of a dynamic ATM 1483 subinterface.

- Example

  host1(config-if)#**auto-configure atm1483**

- Use the **no** version to terminate autodetection of the ATM 1483 encapsulation type.

profile atm1483 bulk-config-name

- Use to assign the base profile configured for a dynamic ATM 1483 subinterface to the VC range configured on a static ATM AAL5 interface.
- You must include both of the following:
  - Name assigned to the VC range on an ATM AAL5 interface, as specified in the **atm bulk-config** command
  - Name assigned to the base profile for a dynamic ATM 1483 subinterface
- Example

  host1(config-if)#**profile atm1483 bulk-config-name test1 test1BaseProfile**

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

profile atm1483 bulk-config-name pvc

- Use to assign an overriding profile to a single ATM PVC that exists within a bulk-configured VC subrange.
- An overriding profile typically includes debugging attributes that help you troubleshoot problems with the ATM 1483 dynamic subinterface column created on the specified PVC.
- The VPI and VCI values of the PVC you specify must exist between the starting VPI/VCI values and ending VPI/VCI values of a VC subrange previously configured with the **atm bulk-config** command.
- Example 1—In this example, a previously configured VC range named test1 includes a VC subrange with VPIs 3–5 and VCIs 201–3200. The following command assigns an overriding profile (test1DebugProfile) to the ATM PVC with VPI 4 and VCI 301 that is within this subrange.

  host1(config-if)#**profile atm1483 bulk-config-name test1 pvc 4 301 test1DebugProfile**

- Example 2—Removes the overriding profile assignment from the ATM PVC with VPI 4 and VCI 301, and restores the original base profile assignment

  host1(config-if)#**no profile atm1483 bulk-config-name test1 pvc 4 301**

- Use the **no** version to remove the overriding profile assignment for the PVC and restore the original base profile assignment.

shutdown

- Use to disable an interface.
- When you disable a dynamic ATM 1483 interface, the **shutdown** command deletes the ATM 1483 dynamic subinterface column and removes the dynamic ATM 1483 subinterface.

- Example

  host1(config-subif)#**shutdown**

- Because the **shutdown** command removes the dynamic ATM 1483 subinterface from the router, issuing a subsequent **no** version of this command has no effect; that is, it does not restart the disabled subinterface.

Changing VC Subranges

Changing VC subranges within a bulk-configured VC range includes the following tasks:

- Adding new VC subranges to an existing VC range
- Removing VC subranges from an existing VC range
- Modifying a VC subrange by shortening or expanding the subrange values
- Merging multiple VC subranges belonging to an existing VC range
- Changing the administrative state of VC subranges

The following sections describe how to perform these tasks.

Adding VC Subranges

You can add a new VC subrange to an existing VC range only when the new subrange does not overlap with any existing subrange. Any overlap causes the addition to fail.

You can add multiple subranges to an existing VC range simultaneously. However, the entire operation fails if even one of the new subranges overlaps with an existing subrange.

The following example specifies the original VC subranges.

host1(config-if)#**atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range 5 5 501 550 vc-range 3 3 301 350**

To add subranges to this bulk-configured VC range, you can choose either of the following methods. Each method adds a new subrange (4, 4, 401, 450) to the existing VC range, test.

- Specify one new subrange at a time.

  host1(config-if)#**atm bulk-config test vc-range 4 4 401 450**

- Specify the new subrange and all the existing subranges. If you use this method, all the existing subranges and their order must match exactly, or the operation fails.

  host1(config-if)#**atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250 vc-range 5 5 501 550 vc-range 3 3 301 350 vc-range 4 4 401 450**

The following operation fails because the order of subranges does not match the existing order.

```
host1(config-if)#atm bulk-config test vc-range 2 2 201 250 vc-range 1 1 101 150
vc-range 5 5 501 550 vc-range 3 3 301 350 vc-range 4 4 401 450
vc-range 6 6 601 650
```

You can create a placeholder VC range by specifying a VC range name without specifying any subrange parameters. This VC range has no circuit reservation, but you can assign a profile to it, and add subranges later as desired. The following commands illustrate this approach.

```
host1(config-if)#atm bulk-config test
host1(config-if)#profile atm1483 bulk-config-name test atmProfile
host1(config-if)#atm bulk-config test vc-range 4 4 401 450 vc-range 6 6 601 650
```

Removing VC Subranges

You can remove VC subranges from an existing VC range if no dynamic ATM 1483 subinterfaces currently exists for any circuit within those subranges. The removal operation fails if any such dynamic ATM 1483 subinterface exists. You must first remove the dynamic ATM 1483 subinterfaces before you can remove the subranges. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.

You can remove only a single specific VC subrange at a time. The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250
vc-range 5 5 501 550 vc-range 3 3 301 350
```

The following command removes one subrange (1, 1, 101, 150) and leaves the remaining subranges, and the named VC range, test, intact.

```
host1(config-if)#no atm bulk-config test vc-range 1 1 101 150
```

To remove more than one VC subrange, you must issue multiple removal commands, one for each subrange. You cannot remove only part of a subrange. A removal command cannot encompass more than one subrange, even if the subranges are adjacent. However, if you do not specify any subranges, you can remove all subranges in the VC, and the named VC range, at the same time.

```
host1(config-if)#no atm bulk-config test
```

Modifying VC Subranges

You can shorten or expand a subrange by modifying the subrange values of a VC range. You can expand a subrange if none of the circuits added overlap with any other subrange. You can shorten a subrange if none of the circuits dropped have existing dynamic ATM 1483 subinterfaces.

You can modify only a single specific subrange at a time. The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250
vc-range 5 5 501 550 vc-range 3 3 301 350
```

The following command modifies the second subrange from (2, 2, 201, 250) to (2, 3, 210, 230).

```
host1(config-if)#atm bulk-config test modify vc-range 2 3 210 230
```

The router retains any overriding profiles assigned to a subrange after you modify the subrange if the override assignment still falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment.

You cannot modify a subrange at the same time you are adding or removing a subrange. If the new modified values for a subrange partially overlap with another subrange, the operation fails and the router displays an error message.

Merging VC Subranges

You can merge multiple subranges of any particular VC range to form a single unified subrange, conserving subrange resources. Merging takes place only when you modify a subrange so that it completely includes at least one other subrange of the same VC range. The merged subranges do not need to be adjacent to each other.

If the encompassing subrange has any circuits that are outside the subranges to be merged, those circuits are added. The encompassing subrange must cover a subrange completely to incorporate it in the merged subrange. The merge operation fails if the encompassing subrange completely overlaps some subranges but only partially overlaps with another subrange. The encompassing subrange does not have to encompass all subranges of the VC range.

Each subrange that is merged with another frees up a subrange. E-series routers currently support a maximum of 300 bulk-configured VC ranges per chassis. Therefore, if a VC range consists of 5 subranges, 295 subranges are still available for subsequent configuration. If you merge 2 of those subranges, resulting in a new total of 4 subranges in the VC range, then 296 subranges are available for configuration.

The router retains any overriding profile assignments on the subranges made before the merger, and applies them to the new merged subrange. You can separate merged subranges either by removing the merged subrange and then adding new separate subranges or by modifying the merged subrange to remove some portion of the subrange and then adding a new subrange.

The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250
vc-range 5 5 501 550 vc-range 3 3 301 350
```

The following command merges two subranges, (1, 1, 101, 150) and (2, 2, 201, 250), and effectively replaces them with the new subrange (1, 2, 101, 250).

```
host1(config-if)#atm bulk-config test modify vc-range 1 2 101 250
```

To separate the merged subranges, you can modify the unified subrange and add subranges as needed, provided that no dynamic ATM 1483 subinterfaces currently exist for any circuit within those subranges.

If you merge subranges by using SNMP, the new merged subrange takes the lowest instance value of the incorporated subranges. For example, if a VC range has three subranges with instance values of 2, 4, and 5 and the subranges with instance values of 2 and 5 are merged, the new merged subrange has an instance value of 2.

Changing the Administrative State of VC Subranges

VC subranges have an administrative state that enables you to remove dynamic ATM 1483 subinterfaces on various subranges that belong to a single VC range. This functionality is important because subrange removal requires that no dynamic ATM 1483 subinterfaces exist for any circuit on that subrange. The removal operation fails if any such interfaces exist.

By default, the administrative state of a VC subrange is up. When you change the administrative state to down by using the **atm bulk-config shutdown** command, the router deletes all dynamic ATM 1483 subinterfaces on the affected subranges. You can use the **show atm subinterface** command or the **show atm vc** command to monitor the progress of the removal of all dynamic ATM 1483 subinterfaces for the specified subrange.

No additional dynamic ATM 1483 subinterfaces can be created for the subrange until you restore the administrative state to up by using the **no atm bulk-config shutdown** command.

The following example specifies the original VC subranges.

```
host1(config-if)#atm bulk-config test vc-range 1 1 101 150 vc-range 2 2 201 250
vc-range 5 5 501 550 vc-range 3 3 301 350
```

You cannot specify a partial subrange; the specified subrange must exactly match a subrange that has already been configured. The following command changes the administrative state of the second subrange (2, 2, 201, 250) to down. The router removes all dynamic interface columns built on any of the circuits in this subrange. No additional dynamic ATM 1483 subinterfaces can be created until you change the administrative state to up.

```
host1(config-if)#atm bulk-config test shutdown vc-range 2 2 201 250
```

The following command changes the administrative state of this same VC subrange to up.

```
host1(config-if)#no atm bulk-config test shutdown vc-range 2 2 201 250
```

You can change the administrative state of all subranges in a VC range at the same time by issuing the command without specifying any subranges. The following command shuts down all four subranges belonging to the named VC range, test, regardless of their current state.

```
host1(config-if)#atm bulk-config test shutdown
```

The time required for the router to complete an administrative state change depends on the number of VC subranges configured.

atm bulk-config

- Use to create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.
- For detailed information about how to use this command, see **atm bulk-config** on page 554.
- Example

  host1(config)#**atm bulk-config test1 vc-range 0 1 101 600**
  vc-range 3 5 201 3200

- Use the **no** version to remove the specified VC range from the ATM AAL5 interface, to remove the specified subranges from the specified VC range, or to remove all subranges from the specified VC range. The **no** version also removes any overriding profile assignments for ATM PVCs within the deleted VC range or VC subrange.

atm bulk-config modify

- Use to expand or shorten the range of the specified VC subrange. You can modify only a single specific subrange at a time.
- You can expand a subrange if none of the added circuits overlap with any other subrange. You can shorten a subrange if none of the dropped circuits have existing dynamic ATM 1483 subinterfaces.
- Modifying a subrange so that it completely includes at least one other subrange from within the same VC range effectively merges the subranges. Each subrange that is merged with another frees up a subrange for subsequent configuration. The subranges that are merged do not need to be adjacent to each other.
- The router retains any overriding profiles assigned to a subrange if the assignment falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment. If two subranges are merged, the router retains overriding profiles that were assigned to the separate subranges and applies the overriding profiles to the newly merged subrange.
- Example

  host1(config-if)#**atm bulk-config test modify vc-range 2 3 210 230**

- There is no **no** version.

atm bulk-config shutdown

- Use to administratively disable (shut down) a specified VC subrange or all subranges in a VC range. The administrative state of a VC subrange is enabled by default.
- Disabling the VC subrange deletes all dynamic ATM 1483 subinterfaces on the affected subranges. You can use the **show atm subinterface** command or the **show atm vc** command to monitor the progress of the removal of all dynamic ATM 1483 subinterfaces for the specified subrange.

- No dynamic ATM 1483 subinterfaces can subsequently be created for the subrange until you restore the administrative state to enabled by using the **no atm bulk-config shutdown** command.
- Example

  host1(config-if)#**atm bulk-config test shutdown vc-range 2 2 201 250**

- Use the **no** version to enable the specified VC subrange or all subranges in a VC range.

Configuring Static ATM Interfaces Within VC Subranges

You can do either of the following on an E-series router:

- Create a static ATM interface within an existing bulk-configured VC subrange
- Create a bulk-configured VC subrange that includes an existing static ATM interface

The following sections describe how to perform these tasks.

Creating Static ATM Interfaces Within VC Subranges

You can configure a static ATM interface with an ATM PVC whose VPI and VCI addresses fall within an existing bulk-configured VC subrange.

To create a static ATM interface within a VC subrange:

1. Create a bulk-configured VC range that includes one or more VC subranges.

   host1(config)#**interface atm 0/0**
   host1(config-if)#**atm bulk-config test vc-range 1 3 32 1031**

2. Specify a static ATM 1483 subinterface.

   host1(config-if)#**interface atm 0/0.2100**

3. Configure an ATM PVC with VPI and VCI values that fall within the bulk-configured VC subrange. In this example, the VPI value (2) is within the VPI range 1–3, and the VCI value (100) is within the VCI range 32–1031.

   host1(config-subif)#**atm pvc 2100 2 100 aal0**

4. Configure the static ATM interface. For example, the **mpls-relay** command creates a ATM layer 2 services over MPLS tunnel on the circuit.

   host1(config-subif)#**mpls-relay 192.168.0.1 2100**

Creating VC Subranges That Include Static ATM Interfaces

You can configure a bulk-configured VC subrange that includes the VPI and VCI addresses belonging to an existing ATM PVC on a static ATM interface. This example is essentially the reverse of the procedure in *Creating Static ATM Interfaces Within VC Subranges* on page 568.

To create a VC subrange that includes a static ATM interface:

1. Specify a static ATM 1483 subinterface.

   host1(config-if)#**interface atm 3/1.201**

2. Configure an ATM PVC on the static ATM 1483 subinterface. In this example, the VPI value is 1 and the VCI value is 101.

   host1(config-subif)#**atm pvc 201 1 101 aal0**

3. Configure the static ATM interface. For example, the **mpls-relay** command creates an ATM layer 2 services over MPLS tunnel on the circuit.

   host1(config-subif)#**mpls-relay 5.1.1.1 201**

4. Create a bulk-configured VC range that includes the VPI and VCI values of the previously configured ATM PVC. In this example, the VPI range (0–2) includes VPI 1, and the VCI range (100–250) includes VCI 101.

   host1(config)#**interface atm 3/1**
   host1(config-if)#**atm bulk-config test2 vc-range 0 2 100 250**

atm bulk-config

- Use to create a bulk-configured VC range on a static ATM AAL5 interface for use by a dynamic ATM 1483 subinterface.
- For detailed information about how to use this command, see **atm bulk-config** on page 554.
- Example

  host1(config)#**atm bulk-config test1 vc-range 0 1 101 600**
  vc-range 3 5 201 3200

- Use the **no** version to remove the specified VC range from the ATM AAL5 interface, to remove the specified subranges from the specified VC range, or to remove all subranges from the specified VC range. The **no** version also removes any overriding profile assignments for ATM PVCs within the deleted VC range or VC subrange.

atm pvc

- Use to configure a PVC on an ATM interface.
- Specify the VCD, the VPI, the VCI, and the encapsulation type. For more information about these parameters, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- You can create a PVC within an existing bulk-configured VC subrange, or a bulk-configured VC subrange that includes the VPI and VCI values of an existing PVC.
- Use the **aal0** encapsulation keyword to cause the router to receive raw ATM cells on this circuit and to forward the cells without performing AAL5 packet reassembly.

- Example

  host1(config-subif)#**atm pvc 10 100 22 aal0**

- Use the **no** version to remove the specified PVC.

interface atm

- Use to select an ATM interface or ATM 1483 subinterface.
- For information about specifying the ATM interface or subinterface, see **interface atm** on page 557.
- Examples

  host1(config)#**interface atm 5/0.1**
  host1(config)#**interface atm 4/0/2**

- Use the **no** version to remove the interface or subinterface.

mpls-relay

- Use to route layer 2 traffic to the specified router.
- For detailed information about using the **mpls-relay** command, see *JUNOSe BGP and MPLS Configuration Guide, Chapter 5, Configuring Layer 2 Services over MPLS*.
- Example

  host1(config-if)#**mpls-relay 10.10.100.2 45**

- Use the **no** version to negate this command.

Configuring VLAN Dynamic Subinterfaces

E-series routers support configuration of dynamic VLAN subinterfaces over static VLAN major interfaces over Ethernet.

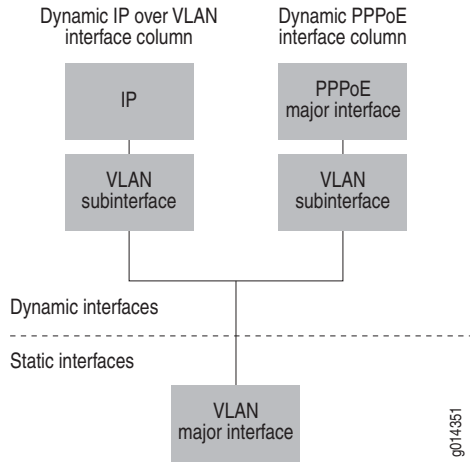
When you configure the dynamic VLAN subinterface, you can enable autodetection and dynamic creation of the following upper-layer encapsulation types:

- IP
- PPPoE

NOTE: Unlike ATM, which supports dynamic upper interfaces over static ATM 1483 subinterfaces, you must configure a dynamic VLAN subinterface to enable autodetection and dynamic creation of IP and PPPoE interfaces.

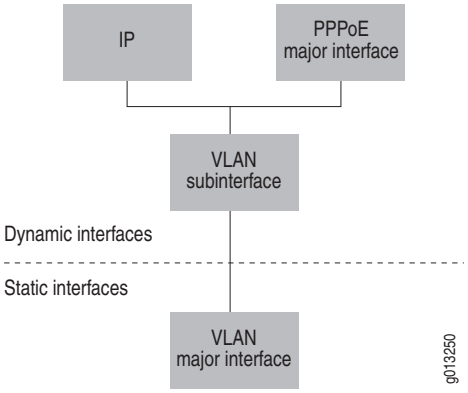
Figure 51 shows the dynamic upper-interface columns supported by dynamic VLAN subinterfaces, and indicates which layers in the columns are static and dynamic.

Figure 51: Dynamic Interface Columns over Dynamic VLAN Subinterfaces



Unlike ATM 1483, you can configure both IP and PPPoE over a single dynamic VLAN subinterface (Figure 52).

Figure 52: Dynamic IP and PPPoE over Single Dynamic VLAN Subinterface



About Configuring Dynamic VLAN Subinterfaces

This section introduces important concepts that you need to understand before you configure dynamic VLAN subinterfaces.

Overview and Benefits

When you configure dynamic VLAN subinterfaces over static VLAN major interfaces, you must configure the VLAN major interface, including the attributes of the VLAN major interface. VLAN major interface attributes include profile assignments and autoconfiguration settings.

As part of the configuration process, you create a VLAN base profile, which can optionally include nested profile assignments, to define the attributes required to configure the dynamic VLAN subinterface and the dynamic upper-layer encapsulation types built over it.

When the router receives a packet, it examines the packet for a VLAN ID or double-tagged S-VLAN ID. You can also configure the router to further examine the packet for agent-circuit-identifier information. Based on these values and the configuration data received from a profile, the router creates all dynamic layers above the VLAN layer, starting with the lowest dynamic layer. For example, in the case of a dynamic PPPoE interface, the router creates the interfaces in the following order:

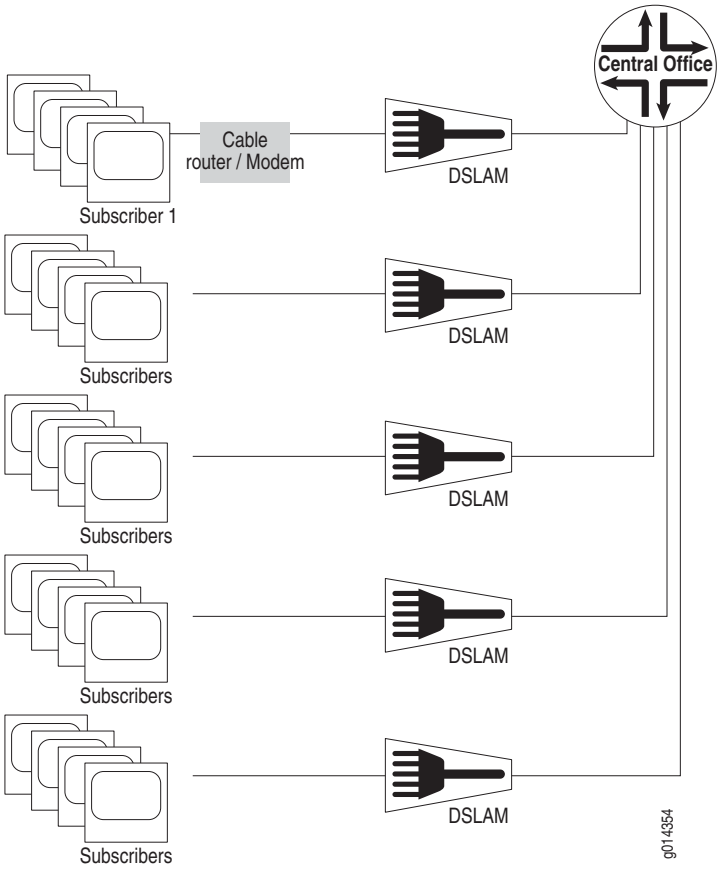
- Dynamic VLAN subinterface
- PPPoE interface
- PPP interface
- IP interface

If any layer of the dynamic portion of the interface column fails to be created, then the interface creation fails and the connection is denied. All dynamic layers above the VLAN subinterface are destroyed, starting with the highest dynamic layer. VLAN subinterfaces are persistent; after they are created, they cannot be destroyed, unless the operational state changes to down.

Dynamic VLAN subinterfaces function identically to static VLAN subinterfaces, except for the manner in which they are created and configured. However, dynamic VLANs provide you with the flexibility of having the dynamic interface column created automatically only when the subscriber logs in.

Figure 53 displays the relationship between the central office, digital subscriber line access multiplexers (DSLAMs), and subscribers. The subscribers are connected to the DSLAMS through Gigabit Ethernet interfaces.

Figure 53: Dynamic VLAN Subinterfaces for Subscribers



For example, if an S-VLAN is assigned at the DSLAM, and each DSLAM subscriber at the DSLAM is assigned a unique VLAN ID, the JUNOSe software dynamically constructs a VLAN-based interface column using that S-VLAN/VLAN ID pair when the subscriber logs in.

For more information about the attributes of VLAN and S-VLAN subinterfaces, see *Chapter 5, Configuring VLAN and S-VLAN Subinterfaces*.

VLAN Base Profiles

To configure a dynamic VLAN subinterface over a static VLAN major interface, you must create a base profile. The base profile includes one or more of the following attributes for the VLAN subinterface:

- advisory-rx-speed—Sets an advisory receive speed for VLAN subinterfaces that are created with this profile. For information, see **vlan advisory-rx-speed** on page 588.
- advisory-tx-speed—Sets an advisory connect speed for VLAN subinterfaces that are created with this profile. For information, see **vlan advisory-tx-speed** on page 588.
- auto-configure—Specifies the types of upper-interface encapsulations that are accepted or detected by the dynamic VLAN subinterface. For information, see **vlan auto-configure** on page 589.
- auto-configure agent-circuit-identifier—Enables the creation of VLAN subinterfaces that are based on agent-circuit-identifier information. For information, see **vlan auto-configure agent-circuit-identifier** on page 589.
- description—Assigns a description to VLAN subinterfaces that are created with this profile. For information, see **vlan description** on page 591.
- policy—Assigns a policy to a VLAN. For information, see **vlan policy** on page 591.
- profile—Adds a nested profile assignment, which references another profile that dynamically configures an upper-interface encapsulation type over the VLAN subinterface. For information, see **vlan profile** on page 592.
- service-profile—Specifies a service profile name for a VLAN. For information, see **vlan service-profile** on page 592.
- svlan ethertype—Specifies that the packet must use this Ethertype to create the dynamic VLAN subinterface. For more information, see **svlan ethertype** on page 588.

You can override the base profile assignment for a VLAN or S-VLAN that exists with a profile. For more information, see *Overriding Base Profile Assignments* on page 578.

Nested Profile Assignments

The configuration for each dynamic upper-interface encapsulation type might differ, depending on the column type built by the router. To manage these differences, you can include one or more nested profile assignments within the VLAN base profile. A nested profile assignment references another profile that configures attributes for a dynamic upper-interface encapsulation type. You can create different profiles for each upper-interface encapsulation type, or you can create a single profile that includes attributes for multiple encapsulation types.

For example, the following commands create a base profile named vlanBaseProfile with two nested profile assignments. The first nested profile assignment references an IP profile named vlanProfileIp, and the second nested profile assignment references a PPPoe profile named vlanProfilePppoe.

```
host1(config)#profile vlanBaseProfile
host1(config-profile)#vlan profile ip vlanProfileIp
host1(config-profile)#vlan profile pppoe vlanProfilePppoe
```

In this example, vlanProfileIp and vlanProfilePppoe have different IP configurations depending on the dynamic interface column constructed. For an IP over VLAN dynamic interface column, the router uses the IP attributes in vlanProfileIp. For an IP over PPPoE dynamic interface column, the router uses the IP attributes in vlanProfilePppoe.

For information about creating profiles for upper-interface encapsulation types, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

Additional Profile Characteristics for Upper Interfaces

In addition to VLAN attributes and nested profile assignments, the base profile for a dynamic VLAN subinterface can also include individual characteristics for several upper-interface encapsulation types, provided that no nested profile assignment for the specified encapsulation type is in the base profile. If, on the other hand, a nested profile assignment for this encapsulation type exists in the base profile, the router obtains all characteristics for that encapsulation type from the nested profile and not from the base profile.

For lists of the characteristics for each supported upper-interface encapsulation type, see *Monitoring Dynamic Interfaces and Profiles* on page 601.

Bulk Configuration of VLAN Ranges

Dynamic creation of VLAN subinterfaces requires you to configure a range of single-tagged VLAN IDs and double-tagged S-VLAN IDs on the VLAN major interface and assign a name to this range. You can also configure a range of S-VLAN IDs that is based on agent-circuit-identifier information. See *Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information* on page 576 for information.

Each VLAN range consists of one or more nonoverlapping VLAN subranges. A VLAN subrange is a group of VLAN IDs and S-VLAN IDs that reside within the VLAN range you specify.

The process of configuring a VLAN range for a dynamic VLAN subinterface is referred to as *bulk configuration*. You create a bulk configuration by issuing the **vlan bulk-config** command. For example, the following commands create a VLAN bulk configuration named myBulkConfig on the specified VLAN interface.

```
host1(config)#interface gigabitEthernet 2/0
host1(config-if)#vlan bulk-config myBulkConfig svlan-range 101 1100 1 375
svlan-range 1300 1500 500 650
```

In the example, the **vlan bulk-config** command configures a VLAN range made up of two VLAN subranges. The first subrange configures S-VLANs 101–1100 and VLANs 1–375. The second subrange configures S-VLANs 1300–1500 and VLANs 500–650.

NOTE: For information about the maximum number of VLAN bulk configurations supported per router and line module, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

After you issue the **vlan bulk-config** command, the router provisions all VLAN IDs and S-VLAN IDs in the specified VLAN range at the same time. The router does not dynamically create the VLAN subinterface until it receives incoming data traffic on the VLAN ID or S-VLAN ID.

After you create a named VLAN range, you cannot remove the underlying VLAN major interface until you issue the **no vlan bulk-config** command to remove the VLAN range from that interface.

Bulk Configuration of VLAN Ranges Using Agent-Circuit-Identifier Information

Using bulk configuration to create S-VLAN IDs based on agent-circuit-identifier information is similar to the process of creating a bulk-configured VLAN range that is not based on agent-circuit-identifier information. However, when you issue the **vlan bulk-config** command with the **svlan-range** keyword to specify the S-VLAN ID range, you then specify the **agent-circuit-identifier** keyword instead of a VLAN ID range. This technique creates a unique type of S-VLAN range in which the agent-circuit-identifier information is used in place of the second tag.

The agent-circuit-identifier string is contained in the option 82 field of DHCP messages for DHCP traffic, or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets for PPPoE traffic. The agent-circuit-identifier information identifies the subscriber's access node and the DSL line on the access node. You can repeat the **svlan-range** and **agent-circuit-identifier** keywords to provide nonoverlapping VLAN subranges that reside within the VLAN range.

The following example configures a VLAN ID range made up of two subranges. The first subrange configures S-VLANs 200–250 and the second subrange configures S-VLANs 3000–3500. Both subranges configure the subscriber identification based on agent-circuit-identifier information.

```
host1(config)#interface gigabitEthernet 2/0
host1(config-if)#vlan bulk-config myAgent2BulkConfig svlan-range 200 250
agent-circuit-identifier svlan-range 3000 3500 agent-circuit-identifier
```

After you issue the **vlan bulk-config** command with the **agent-circuit-identifier** keyword, the router provisions the S-VLAN IDs in the specified bulk-configured VLAN range at the same time. The router does not dynamically create the VLAN subinterface until it receives incoming data traffic. The user information is generated from the incoming data traffic that contains the agent-circuit-identifier string.

Conceptually, a VLAN subinterface in this configuration has two attributes, an S-VLAN ID and an agent-circuit-identifier string. This is analogous to a regular S-VLAN that also has two attributes, an S-VLAN ID and a VLAN ID. However, the packet that the router receives is singly-tagged with only a VLAN ID. The use of the **agent-circuit-identifier** keyword in the **vlan bulk-config** command causes the router to further examine the packet and extract the agent-circuit-identifier string in order to generate the subscriber identification information.

In a DSL access network, subscriber information can be conveyed through either of the following methods:

- VLAN encapsulation; that is, the S-VLAN ID and the VLAN ID
- Insertion of the agent-circuit-identifier string in DHCP or PPPoE messages

For example, the following configurations uniquely identify subscribers by means of VLAN encapsulation:

- Subscriber packets received from the DSLAM are single-tagged with a VLAN ID
- Subscriber packets received from the DSLAM are double-tagged with both an S-VLAN ID and a VLAN ID

The DSL Forum Technical Report (TR)-101—Migration to Ethernet-Based DSL Aggregation (April 2006) refers to the behavior of these configurations as the 1:1 forwarding model because there is a one-to-one correspondence between an individual subscriber and the VLAN encapsulation.

In contrast, the following configurations do *not* uniquely identify subscribers by means of VLAN encapsulation:

- Subscriber packets received from the DSLAM are single-tagged with the same VLAN ID for a group of subscribers. This configuration is typically used to implement service VLANs where the VLAN ID corresponds to the type of service for which the VLAN is used, such as voice or video. In this configuration, the VLAN ID does not correspond to an individual subscriber.
- Subscriber packets received from the DSLAM are untagged.

Instead, these configurations identify subscribers by means of the agent-circuit-identifier information present in DHCP and PPPoE control messages. DSL Forum TR-101 refers to the behavior of these configurations as the N:1 forwarding model because there is a many-to-one correspondence between subscribers and a VLAN.

Creating dynamic VLANs based on agent-circuit-identifier information enables you to manage subscribers in single-tagged or untagged N:1 configurations that do not use encapsulation to uniquely identify subscribers. In these configurations, the router intercepts the agent-circuit-identifier string from DHCP messages or from PPPoE PADR and PADI packets to build a unique subscriber interface.

For double-tagged 1:1 configurations, the router uses standard dynamic VLAN procedures to uniquely identify subscribers. In these configurations, the S-VLAN ID typically represents the DSLAM, and the VLAN ID represents the individual subscriber accessing the router through that DSLAM.

For configuration instructions, see *Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information* on page 581.

NOTE: You must configure the DHCP local or external server to support the creation of dynamic subscriber interfaces that are based on the agent-circuit-id option (suboption 1) of the option 82 field in DHCP messages. See *JUNOSe Broadband Access Configuration Guide, Chapter 19, Configuring DHCP Local Server* or *JUNOSe Broadband Access Configuration Guide, Chapter 21, Configuring the DHCP External Server Application* for information.

Dynamic Interface Creation

After you configure the base profile, you associate it with the VLAN major interface by issuing the **profile vlan bulk-config** command.

As a final step, you must issue the **auto-configure vlan** command. This command configures the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type.

When the router receives an incoming data packet on a circuit, it dynamically creates the VLAN subinterface, using the attributes specified in the base profile. After examining the contents of the data packet, the router dynamically creates the required interface columns above the VLAN subinterface, using the configuration attributes contained in the nested profiles, if specified, or in the base profile itself.

Overriding Base Profile Assignments

You can also use the **profile vlan override bulk-config** command to assign an overriding profile to a single VLAN ID or double-tagged S-VLAN ID that exists within a bulk-configured VLAN subrange. The VLAN ID subrange that encompasses the major interface must have been previously configured with the **vlan bulk-config** command for use by a dynamic VLAN subinterface. After you assign the overriding profile, the router uses the information in this profile instead of the information in the previously assigned base profile to create any subsequent VLAN dynamic subinterface columns on the specified VLAN major interface, as long as they match the VLAN or S-VLAN specified in the override.

The overriding profile, like the original base profile, can include VLAN attributes, nested profile assignments, and individual characteristics for dynamic upper-interface encapsulation types.

Overriding the base profile assignment for a VLAN with a profile enables you to create a special profile for a subscriber in a DSLAM. For example, you can use the overriding profile to create dynamic VLAN subinterfaces for subscribers with an S-VLAN ID of 200 and a VLAN ID of 100.

You can also use an overriding profile with debugging attributes to troubleshoot problems with VLAN dynamic subinterface columns.

For configuration instructions and examples, see *Configuring Overriding Profile Assignments for VLAN Major Interfaces* on page 582.

NOTE: See *JUNOSe Release Notes*, *Appendix A*, *System Maximums* for information about the maximum number of overriding profile assignments currently supported per chassis.

Changing VLAN Subranges

You can add, remove, modify, merge, disable, and enable VLAN subranges within an existing bulk-configured VLAN range.

For configuration instructions and examples, see *Changing VLAN Subranges* on page 592.

Static VLAN Subinterfaces Within VLAN Subranges

You can configure a static VLAN subinterface with a single-tagged VLAN ID or double-tagged S-VLAN ID, or an S-VLAN ID with agent-circuit-identifier information that falls within an existing bulk-configured VLAN subrange. Conversely, you can also create a bulk-configured VLAN subrange that includes the single-tagged VLAN ID or double-tagged S-VLAN ID on a static VLAN subinterface. Configuring static VLAN subinterfaces within VLAN subranges can be useful when you want to create a column statically for users who have difficulty logging on. You might also want to configure static VLAN subinterface within a VLAN subrange as a static column to the DSLAM; the dynamic column can be for subscribers.

The following rules apply when you configure either a static VLAN subinterface within an existing bulk-configured VLAN subrange or a subrange that includes an existing static VLAN interface:

- You have no restrictions on how to configure the static VLAN subinterface.
- Static VLAN interfaces defined within a bulk-configured VLAN subrange are stored in NVS and preserved after a reboot.
- The base profile associated with the VLAN subrange does not apply to any statically defined VLAN interfaces that fall within the subrange.
- If a VLAN subrange includes a statically defined VLAN subinterface, overriding profile assignments configured for the same VLAN ID as a statically defined VLAN subinterface become inactive until the static VLAN subinterface is removed. The overriding profile becomes active again when you remove the static VLAN subinterface. You can display the current operational status (active or inactive) of overriding profile assignments by using the **show vlan bulk-config** command.

- Operations that add, remove, modify, merge, disable, or enable VLAN subranges within a bulk-configured VLAN range do not affect any static VLAN subinterfaces defined within the VLAN subrange.

- You cannot create a static VLAN if the single-tagged VLAN ID or double-tagged S-VLAN ID conflicts with an existing VLAN dynamic subinterface column. Such a configuration would disrupt subscribers already connected to the router via the dynamic subinterface.

For configuration information and examples, see *Configuring Static VLAN Subinterfaces Within VLAN Subranges* on page 598.

Configuring a Dynamic VLAN Subinterface

To configure a dynamic VLAN subinterface:

1. Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic VLAN subinterface.

   These profiles are referenced in the base profile for the dynamic VLAN subinterface as nested profile assignments. For detailed instructions on creating profiles, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

2. (Optional) Create the profile for an upper-interface encapsulation type, and include additional profile characteristics for other encapsulation types as needed. Perform this step if you want to create a nested profile assignment in Step 5.

   ```
   host1(config)#profile myIpProfile
   host1(config-profile)#ip inactivity-timer 200
   host1(config-profile)#ip auto-configure ip-subscriber include-primary
   ```

3. Create the base profile for the dynamic VLAN subinterface by assigning the profile a name.

   ```
   host1(config)#profile vlanBaseProfile
   ```

   This command accesses Profile Configuration mode, which enables you to configure attributes in the base profile.

4. Define attributes for the VLAN subinterface in the base profile.

   a. Configure the VLAN major interface for autodetection of the PPPoE upper-interface encapsulation type.

   b. Configure the VLAN subinterface for autodetection of the IP upper-interface encapsulation type.

   c. Configure an Ethertype value for any S-VLANs configured on the VLAN.

   ```
   host1(config-profile)#vlan auto-configure pppoe
   host1(config-profile)#vlan auto-configure ip
   host1(config-profile)#svlan ethertype 8100
   ```

5. (Optional) In the base profile, create nested profile assignments for the upper-interface encapsulation types.

   For example, the following command configures nested profile assignments for the IP upper-interface encapsulation types.

   host1(config-profile)#**vlan profile ip myIpProfile**

6. Exit Profile Configuration mode.

7. Configure the VLAN major interface.

   host1(config)#**interface gigabitEthernet 5/0**
   host1(config-if)#**encapsulation vlan**

8. Configure a VLAN range on the major VLAN interface, and assign a name to this range.

NOTE: For information about the maximum number of VLAN bulk configurations supported per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

   For example, the following command creates a VLAN range named myBulkConfig made up of two VLAN subranges.

   host1(config-if)#**vlan bulk-config myBulkConfig vlan-range 0 100 vlan-range 110 200**

9. Assign the base profile configured for the VLAN subinterface to the VLAN range configured on the major VLAN interface.

   host1(config-if)#**profile vlan bulk-config myBulkConfig vlanBaseProfile**

10. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type.

    host1(config-if)#**auto-configure vlan**

Configuring Dynamic VLAN Subinterfaces Based on Agent Circuit Identifier Information

The procedure you use to configure a dynamic VLAN subinterface that is based on agent-circuit-identification information is similar to the procedure described in *Configuring a Dynamic VLAN Subinterface* on page 580.

1. Configure profiles containing characteristics for the dynamic upper-interface encapsulation types to be created over the dynamic VLAN subinterface.

2. (Optional) If you want to create a nested profile assignment, create the profile for an upper-interface encapsulation type, and include additional profile characteristics for other encapsulation types as needed.

3. Create the base profile for the dynamic VLAN subinterface and enter Profile Configuration mode by assigning the profile a name.

   host1(config)#**profile vlanMyBaseProfile**

4. Define attributes for the VLAN subinterface in the base profile.

   a. Enable autoconfiguration for the PPPoE upper-interface encapsulation type.

   b. Enable autoconfiguration for the IP upper-interface encapsulation type.

   c. Enable autoconfiguration of VLANs that are based on agent-circuit-identifier information.

   d. (Optional) Create nested profile assignments for the upper-interface encapsulation types.

   host1(config-profile)#**vlan auto-configure pppoe**
   host1(config-profile)#**vlan auto-configure ip**
   host1(config-profile)#**vlan auto-configure agent-circuit-identifier**
   host1(config-profile)#**exit**
   host1(config)#

5. Configure the VLAN major interface.

   host1(config)#**interface gigabitEthernet 5/0**
   host1(config-if)#**encapsulation vlan**

6. On the VLAN major interface, configure a VLAN range that is based on agent-circuit-identifier information, and assign a name to this range.

   host1(config-if)#**vlan bulk-config myNewBulkConfig svlan-range 50 100 agent-circuit-identifier**

7. Assign the base profile configured for the VLAN subinterface to the VLAN range configured on the major VLAN interface.

   host1(config-if)#**profile vlan bulk-config myNewBulkConfig vlanMyBaseProfile**

8. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type.

   host1(config-if)#**auto-configure vlan**

Configuring Overriding Profile Assignments for VLAN Major Interfaces

You can assign an overriding profile to a single VLAN major interface within a bulk-configured VLAN subrange.

The overriding profile includes debugging attributes to help you identify and troubleshoot problems with the VLAN dynamic subinterface column created on the specified VLAN ID.

To assign an overriding profile to a VLAN within a bulk-configured VLAN subrange:

1. Configure both of the following:

   - Base profile for the bulk-configured dynamic VLAN on the static VLAN major interface. The VLAN range consists of one or more VLAN subranges.

   - Overriding profile for a dynamic VLAN within a bulk-configured VLAN subrange.

   For information about configuring profiles, see *Configuring a Dynamic Interface from a Profile* in *Chapter 15, Configuring Dynamic Interfaces*.

2. Create a bulk-configured range of single-tagged VLAN IDs or double-tagged S-VLAN IDs on a static VLAN major interface. The following commands create a bulk-configured VLAN range named myBulkConfig that consists of two VLAN subranges. The first subrange encompasses VLAN IDs 150–250. The second subrange encompasses VLAN IDs 300–500.

   host1(config)#**interface gigabitEthernet 4/0.101**
   host1(config-if)#**vlan bulk-config myBulkConfig vlan-range 150 250 vlan-range 300 500**

3. Assign the previously configured base profile (vlanBaseProfile) to the bulk-configured VLAN range.

   host1(config-if)#**profile vlan bulk-config myBulkConfig vlanBaseProfile**

4. Assign the previously configured overriding profile to a single VLAN ID or double-tagged S-VLAN ID within the bulk-configured VLAN subrange. The following command assigns the overriding profile overrideVoiceSubscriber to the VLAN ID 202. This VLAN ID exists within the first VLAN subrange (VLAN IDs 150–250) configured in Step 2.

   host1(config-if)#**profile vlan override bulk-config myBulkConfig vlan 202 overrideVoiceSubscriber**

   The router now uses the information in the overriding profile instead of the information in the base profile to create subsequent VLAN dynamic subinterface columns over this VLAN ID.

5. (Optional) You can assign the same overriding profile to a VLAN ID within the same VLAN range or within a different VLAN range. For example, the following command assigns the overriding profile overrideVoiceSubscriber to the VLAN ID 160. This S-VLAN ID exists within the VLAN subrange configured in Step 2.

   host1(config-if)#**profile vlan override bulk-config-name myBulkConfig svlan 120 202 overrideVoiceSubscriber**

NOTE: You can reverse the order of Step 2 and Step 4 with identical results. That is, you can assign the overriding profile to an S-VLAN ID and then assign the base profile to the entire VLAN subinterface.

6. Configure the VLAN major interface to support autodetection of the VLAN dynamic encapsulation type.

   host1(config-if)#**auto-configure vlan**

7. (Optional) Use the **show vlan profile** command to verify the overriding profile configuration.

   For more information about using this command, see *Monitoring Dynamic Interfaces and Profiles* on page 601.

Removing an Overriding Profile Assignment from a VLAN

You can remove an overriding profile assignment from a VLAN major interface.

If you use the overriding profile to troubleshoot the VLAN dynamic subinterface column created on the specified VLAN ID, make sure that you remove the overriding profile assignment to restore the original base profile assignment. This action ensures that subsequent VLAN dynamic subinterface columns are created using the same attributes defined in the base profile.

To remove an overriding profile assignment from a VLAN:

1. Remove the overriding profile assignment from the specified VLAN ID or S-VLAN ID.

   host1(config-if)#**no profile vlan override bulk-config-name myBulkConfig vlan 202 overrideVoiceSubscriber**

2. Select the dynamic VLAN subinterface on which the VLAN dynamic subinterface column resides.

   host1(config)#**interface gigabitEthernet 4/0.101**

3. Use the **shutdown** command to disable the dynamic VLAN subinterface. The **shutdown** command deletes the VLAN dynamic subinterface column and removes the dynamic VLAN subinterface.

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

4. Send traffic over the VLAN subinterface. This action re-creates the VLAN dynamic subinterface column with the original base profile association.

   The router now uses the information in the base profile instead of the information in the overriding profile to create subsequent VLAN dynamic subinterface columns for the specified VLAN ID or S-VLAN ID.

5. (Optional) Use the **show vlan profile override** command to verify the removal of the overriding profile assignment.

   For more information about using this command, see *Monitoring Dynamic Interfaces and Profiles* on page 601.

Removing Overriding Profile Assignments from a VLAN Range or VLAN Subrange

When you issue the **no vlan bulk-config** command to remove an entire VLAN range (and all VLAN subranges within that VLAN range), the router also removes any overriding profile assignments configured for VLAN IDs within those VLAN subranges. For example, the following command removes the bulk-configured VLAN range named myBulkConfig and any overriding profile assignments for VLAN IDs within the VLAN subranges belonging to myBulkConfig.

```
host1(config-if)#no vlan bulk-config myBulkConfig
```

When you issue the **no vlan bulk-config** command to remove a particular VLAN subrange in a bulk-configured VLAN range, the router also removes any overriding profile assignments for VLAN IDs within that VLAN subrange. However, overriding profile assignments for VLAN IDs within other VLAN subranges in the VLAN range remain intact. For example, the following command removes one VLAN subrange (S-VLAN IDs 50–150 and VLAN IDs 150–250) and only those overriding profile assignments associated with this subrange.

```
host1(config-if)#no vlan bulk-config myBulkConfig svlan-range 50 150 150 250
```

auto-configure vlan

- Use to configure the static VLAN major interface to support autodetection of an VLAN dynamic interface type.
- You must issue this command to enable creation of a dynamic VLAN subinterface.
- By default, all valid VLAN IDs and S-VLAN IDs are accepted.
- Example

  ```
  host1(config-if)#auto-configure vlan
  ```

- Use the **no** version to terminate autodetection of the VLAN dynamic interface type.

encapsulation vlan

- Use to configure VLAN as the encapsulation method for the interface.
- Example

  ```
  host1(config-if)#encapsulation vlan
  ```

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

interface fastEthernet

- Use to select a Fast Ethernet interface.
- For information about specifying a Fast Ethernet interface, see **interface fastEthernet** on page 585.

- Example

  host1(config)#**interface fastEthernet 4/1**

- Use the **no** version to remove IP from an interface or a subinterface.

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- To specify a Gigabit Ethernet interface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*[.*subinterface*] format.
- To specify a Gigabit Ethernet interface or 10-Gigabit Ethernet interface for E120 and E320 routers, use the *slot*/*adapter*/*port*[.*subinterface*] format.
- For more information, see *JUNOSe Physical Layer Configuration Guide, Chapter 5, Configuring Ethernet Interfaces*.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

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

profile

- Use to create a base profile to configure attributes for a dynamic VLAN subinterface.
- Specify a profile name of up to 80 alphanumeric characters.
- Example

  host1(config)#**profile vlanBaseProfile**

- Use the **no** version to delete the specified profile.

profile vlan bulk-config

- Use to assign the base profile configured for a dynamic VLAN subinterface to the single-tagged VLAN IDs or double-tagged S-VLAN IDs configured on a static VLAN major interface.
- You must specify both of the following:
  - Name assigned to the VLAN range on a VLAN subinterface, as specified in the **vlan bulk-config** command
  - Name assigned to the base profile for a dynamic VLAN subinterface
- Example

  host1(config-if)#**profile vlan bulk-config myBulkConfig vlanBaseProfile**

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

profile vlan override bulk-config

- Use to assign an overriding profile to a single VLAN ID or double-tagged S-VLAN ID.
- Using an overriding profile enables you to assign a special profile for the subscribers associated with a specific DSLAM.
- You can also use an overriding profile to troubleshoot the specified VLAN or S-VLAN by overriding the currently assigned base profile with one that has debugging attributes enabled.
- Use the **any** keyword to specify a VLAN ID as a wildcard. When you specify the **any** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM will be created with the same profile.
- Example 1—Assigns an overriding profile (test1OverridingProfile) to the dynamic VLAN subinterface with VLAN ID 202

  host1(config-if)#**profile vlan override bulk-config vlan 202 test10verridingProfile**

- Example 2—Assigns an overriding profile (test1DebugProfile) to the S-VLAN subinterface with S-VLAN ID 100 within the VLAN subinterface with V-LAN ID 202

  host1(config-if)#**profile vlan override bulk-config svlan 100 202 test10verridingProfile**

- Example 3—Removes the overriding profile assignment from the VLAN subinterface with VLAN ID 202, and restores the original base profile assignment

  host1(config-if)#**no profile vlan override bulk-config vlan 202 test10verridingProfile**

- Use the **no** version to remove the overriding profile assignment for the VLAN ID or S-VLAN ID and restore the original base profile assignment.

shutdown

- Use to disable an interface.
- When you disable a dynamic VLAN subinterface, the **shutdown** command deletes the VLAN dynamic subinterface column and removes the dynamic VLAN subinterface.
- Example

  host1(config-subif)#**shutdown**

- Because the **shutdown** command removes the dynamic VLAN subinterface from the router, issuing a subsequent **no** version of this command has no effect; that is, it does not restart the disabled subinterface.

svlan ethertype

- Use to specify the available Ethertypes that a packet must use to create a dynamic VLAN subinterface.
- Choose one of the following Ethertype values:
  - **8100**—Specifies Ethertype value 0x8100, as defined in IEEE Standard 802.1q
  - **88a8**—Specifies Ethertype value 0x88a8, as defined in draft IEEE Standard 802.1ad
  - **9100**—Specifies Ethertype value 0x9100
  - **autoconfig**—Specifies that the packet can use any Ethertype to create a dynamic VLAN subinterface
- Examples

  host1(config-profile)#**svlan ethertype 8100**
  host1(config-profile)#**svlan ethertype autoconfig**

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

vlan advisory-rx-speed

- Use to set an advisory receive speed for VLAN subinterfaces that are created with the profile you are configuring. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *JUNOSe Broadband Access Configuration Guide, Chapter 12, Configuring an L2TP LAC* for additional information about the advisory receive speed.
- The range is 0–2147483647 kbps; 0 indicates no advisory speed setting.
- Example

  host1(config-profile)#**vlan advisory-rx-speed 2000**

- Use the **no** version to restore the default behavior—the Rx speed is not sent to the LNS.

vlan advisory-tx-speed

- Use to set an advisory connect speed for VLAN subinterfaces that are created with the profile that you are configuring. This setting has no effect on data forwarding. You can use it to indicate the speed of the client interface. When traffic is tunneled with L2TP, the advisory receive speed is sent from the LAC to the LNS. See *JUNOSe Broadband Access Configuration Guide, Chapter 12, Configuring an L2TP LAC* for additional information about the advisory receive speed.
- The range is 0–2147483647 kbps; 0 indicates no advisory speed setting.
- Example

  host1(config-profile)#**vlan advisory-tx-speed 2000**

- Use the **no** version to restore the default behavior—the Tx speed is not sent to the LNS.

vlan auto-configure

- Use to specify the types of dynamic upper-interface encapsulations that are accepted or detected by a dynamic VLAN subinterface.
- Include this command in the base profile for a dynamic VLAN subinterface.
- Use the **lockout-time** keyword to specify the minimum and maximum lockout time range for the encapsulation type. For more information, see *Encapsulation Type Lockout* on page 449.
- Example

  host1(config-profile)#**vlan auto-configure ip**

- Use the **no** version to terminate detection of the specified encapsulation type.

vlan auto-configure agent-circuit-identifier

- Use to create a VLAN subinterface that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets.
- Include this command in the base profile for a dynamic VLAN subinterface.
- Example

  host1(config-profile)#**vlan auto-configure agent-circuit-identifier**

- Use the **no** version to disable creation of VLAN subinterfaces based on agent-circuit-identifier information.

vlan bulk-config

- Use to create a bulk-configured VLAN range on a static VLAN major interface for use by a dynamic VLAN subinterface.

NOTE: For information about the maximum number of VLAN bulk configurations supported per chassis, see *JUNOSe Release Notes*, *Appendix A*, *System Maximums*.

- Each VLAN range consists of one or more nonoverlapping VLAN subranges. A VLAN subrange is a group of VLAN IDs or S-VLAN IDs that reside within the VLAN range you specify.
- You can configure multiple VLAN ranges on a VLAN subinterface.
- When you create a bulk-configured VLAN range, you must specify the following:
  - A name of up to 80 alphanumeric characters; this is also referred to as the bulk configuration name
  - The starting and ending VLAN ID or S-VLAN ID values (inclusive) for each VLAN subrange
- Use the **any** keyword to specify a VLAN ID as a wildcard. When you specify the **any** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM will be created with the same profile.

- Use the **agent-circuit-identifier** keyword to configure a VLAN range that is based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. When you specify the **agent-circuit-identifier** keyword with an S-VLAN ID of a DSLAM, all subscribers associated with the DSLAM are created with the same profile.
- You can create a placeholder VLAN range by issuing the **vlan bulk-config** command without specifying any subranges. You can assign a profile to this placeholder and add subranges to it later.
- You can add and remove individual VLAN subranges.
- You cannot remove a VLAN subrange if any dynamic VLAN subinterfaces currently exist within the subrange. Use the **vlan bulk-config shutdown** command to remove dynamic VLAN interfaces created within a subrange.
- Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.
- You can create a bulk-configured VLAN subrange that includes the VLAN IDs and S-VLAN IDs belonging to an existing VLAN major interface on a static VLAN subinterface.
- Example 1—Configures a VLAN range named myBulkConfig with a single VLAN subrange containing VLAN IDs 100–500

  host1(config-if)#**vlan bulk-config myBulkConfig vlan-range 100 500**

- Example 2—Configures a VLAN range named myMultiBulkConfig with two VLAN subranges containing S-VLAN IDs 101–600 with VLAN IDs 0–1 (first subrange) and S-VLAN IDs 201–3200 with VLAN IDs 3–5 (second subrange)

  host1(config-if)#**vlan bulk-config myMultiBulkConfig svlan-range 101 600 0 1 svlan-range 201 3200 3 5**

- Example 3—Configures a VLAN range named myAciBulkConfig containing S-VLAN IDs 200–400. Subscriber information is determined by the packet's agent-circuit-identifier information.

  host1(config-if)#**vlan bulk-config myAciBulkConfig svlan-range 200 400 agent-circuit-identifier**

- Use the **no** version to remove the specified VLAN range from the VLAN interface, to remove the specified subranges from the specified VLAN range, or to remove all subranges from the specified VLAN range. The **no** version also removes any overriding profile assignments for VLAN major interfaces within the deleted VLAN range or VLAN subrange.

vlan description

- Use to assign a description to VLAN subinterfaces that are created with this profile.
- You can use a maximum of 64 characters for the description or to name the alias.
- Example

  host1(config-profile)#**vlan description test1**

- Use the **no** version to remove the VLAN description.

vlan policy

- Use to assign a VLAN policy list to an interface.
- Use the **input** or **output** keyword to assign the policy list to the ingress or egress of the interface.
- You can enable or disable the recording of routing statistics for bytes and packets affected by the policy.
- If you enable statistics, you can enable or disable baselining of the 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.
- You must also enable baselining on the interface with the appropriate **baseline** command.
- You can use the **preserve** keyword to save the existing statistics when you attach a policy to an interface that already has a policy attached. This keyword saves the statistics for any classifier-list that is the same for both the new and old policy attachments. Without the **preserve** keyword, all statistics are deleted when you attach the new policy.

  For example, when you replace a policy attachment that references the original policy-list plOne with a new attachment referencing policy-list plTwo, the existing statistics for the classifier group referencing clOne and the default classifier group are saved.

Original Policy Attachment	New Policy Attachment	Comment
ip policy-list plOne	ip policy-list plTwo	–
ip classifier-list clOne	ip classifier-list clOne	statistics from plOne are saved
Forward	Forward	–
ip classifier-list clTwo	ip classifier-list clFour	–
Forward	Forward	–
ip classifier-list clThree	ip classifier-list clFive	–
Forward	Forward	–
classifier-list *	classifier-list *	statistics from plOne are saved
Filter	Filter	–

- Example

  host1(config-profile)#**vlan policy input VlanPolicy33 statistics enabled preserve**

- Use the **no** version to remove the association between a policy list and an interface or a profile.

vlan profile

- Use to add a nested profile assignment to a base profile for a dynamic VLAN subinterface.
- A nested profile assignment references another profile that configures attributes for a dynamic upper-interface type over the VLAN subinterface.
- Examples

  host1(config-profile)#**vlan profile pppoe vlanProfilePppoe**
  host1(config-profile)#**vlan profile ip vlanProfileIP**

- Use the **no** version to remove the profile assignment for the upper-interface encapsulation type.

vlan service-profile

- Use to specify a service profile name for a dynamic VLAN and to enter Service Profile Configuration mode. Service profiles contain user and password information, and are used in route maps for subscriber management and to authenticate subscribers with RADIUS.
- You can specify a service profile name with up to 80 alphanumeric characters.
- Example

  host1(config)#**vlan service-profile vlanClass1Service**
  host1(config-service-profile)#

- Use the **no** version to delete the service profile.

Changing VLAN Subranges

Changing VLAN subranges within a bulk-configured VLAN range includes the following tasks:

- Adding new VLAN subranges to an existing VLAN range
- Removing VLAN subranges from an existing VLAN range
- Modifying a VLAN subrange by shortening or expanding the subrange values
- Merging multiple VLAN subranges belonging to an existing VLAN range
- Changing the administrative state of VLAN subranges

The following sections describe how to perform these tasks.

Adding VLAN Subranges

You can add a new VLAN subrange to an existing VLAN range only when the new subrange does not overlap with any existing subrange. Any overlap causes the addition to fail.

You can add multiple subranges to an existing VLAN range simultaneously. However, the entire operation fails if even one of the new subranges overlaps with an existing subrange.

The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 201 250 2 2
svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

To add subranges to this bulk-configured VLAN range, you can choose either of the following methods. Each method adds a new subrange encompassing S-VLAN IDs 401–450 with VLAN ID 4 to the existing VLAN range, test.

- Specify one new subrange at a time.

  ```
  host1(config-if)#vlan bulk-config test svlan-range 401 450 4 4
  ```

- Specify the new subrange and all the existing subranges. If you use this method, all the existing subranges and their order must match exactly, or the operation fails.

  ```
  host1(config-if)#vlan bulk-config test svlan-range 201 250 2 2
  svlan-range 501 550 5 5 svlan-range 301 350 3 3 svlan-range 401 450 4 4
  ```

  The following operation fails because the order of subranges does not match the existing order.

  ```
  host1(config-if)#vlan bulk-config test svlan-range 201 250 2 2
  svlan-range 101 150 1 1 svlan-range 501 550 5 5 svlan-range 301 350 3 3
  svlan-range 401 450 4 4 svlan-range 601 650 6 6
  ```

You can create a placeholder VLAN range by specifying a VLAN range name without specifying any subrange parameters. This VLAN range has no VLAN ID reservation, but you can assign a profile to it, and add subranges later as desired. The following commands illustrate this approach.

```
host1(config-if)#vlan bulk-config test
host1(config-if)#profile vlan bulk-config-name test vlanProfile
host1(config-if)#vlan bulk-config test svlan-range 401 450 4 4
svlan-range 601 650 6 6
```

Removing VLAN Subranges

You can remove VLAN subranges from an existing VLAN range if no dynamic VLAN subinterfaces currently exists for any circuit within those subranges. The removal operation fails if any such dynamic VLAN subinterface exists. You must first remove the dynamic VLAN subinterfaces before you can remove the subranges. Removal of a subrange automatically results in the removal of all overriding profile assignments on that subrange.

You can remove only a single specific VLAN subrange at a time. The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

The following command removes one subrange encompassing S-VLAN IDs 101–150 with VLAN ID 1 and leaves the remaining subranges, and the named VLAN range, test, intact.

```
host1(config-if)#no vlan bulk-config test svlan-range 101 150 1 1
```

The following command removes a subrange that includes S-VLAN IDs 700–750, and that is based on agent-circuit-identifier information from the named VLAN range, test.

```
host1(config-if)#no vlan bulk-config test svlan-range 700 750
agent-circuit-identifier
```

To remove more than one VLAN subrange, you must issue multiple removal commands, one for each subrange. You cannot remove only part of a subrange. A removal command cannot encompass more than one subrange, even if the subranges are adjacent. However, if you do not specify any subranges, you can remove all subranges in the VLAN, and the named VLAN range, at the same time.

```
host1(config-if)#no vlan bulk-config test
```

Modifying VLAN Subranges

You can shorten or expand a subrange by modifying the subrange values of a VLAN range. You can expand a subrange if none of the VLAN IDs or S-VLAN IDs added overlap with any other subrange. You can shorten a subrange if none of the VLAN IDs or S-VLAN IDs have existing dynamic VLAN subinterfaces. You can also modify an existing subrange by configuring it to use agent-circuit-identifier information rather than a range of VLAN IDs.

You can modify only a single specific subrange at a time. The following example specifies the original VLAN subranges encompassing S-VLAN IDs 201–250 with VLAN ID 2.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

The following command modifies the second subrange from S-VLAN IDs 201–250 with VLAN ID 2 to S-VLAN IDs 210–230 with VLAN IDs 2–3.

```
host1(config-if)#vlan bulk-config test modify svlan-range 210 230 2 3
```

The following command modifies the third subrange from S-VLAN IDs 501–550 with VLAN ID 5 to S-VLAN IDs 501–550 with user identification that is based on agent-circuit-identifier information.

```
host1(config-if)#vlan bulk-config test modify svlan-range 501 550
agent-circuit-identifier
```

The router retains any overriding profiles assigned to a subrange after you modify the subrange if the override assignment still falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment.

You cannot modify a subrange at the same time you are adding or removing a subrange. If the new modified values for a subrange partially overlap with another subrange, the operation fails and the router displays an error message.

Merging VLAN Subranges

You can merge multiple subranges of any particular VLAN range to form a single unified subrange, conserving subrange resources. Merging takes place only when you modify a subrange so that it completely includes at least one other subrange of the same VLAN range. The merged subranges do not need to be adjacent to each other.

If the encompassing subrange has any VLAN IDs or S-VLAN IDs that are outside the subranges to be merged, those VLAN IDs or S-VLAN IDs are added. The encompassing subrange must cover a subrange completely to incorporate it in the merged subrange. The merge operation fails if the encompassing subrange completely overlaps some subranges but only partially overlaps with another subrange. The encompassing subrange does not have to encompass all subranges of the VLAN range.

Each subrange that is merged with another frees up a subrange. E-series routers currently support a maximum of 300 bulk-configured VLAN ranges per chassis. Therefore, if a VLAN range consists of 5 subranges, 295 subranges are still available for subsequent configuration. If you merge 2 of those subranges, resulting in a new total of 4 subranges in the VLAN range, then 296 subranges are available for configuration.

The router retains any overriding profile assignments on the subranges made before the merger, and applies them to the new merged subrange. You can separate merged subranges either by removing the merged subrange and then adding new separate subranges or by modifying the merged subrange to remove some portion of the subrange and then adding a new subrange.

The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

The following command merges two subranges (S-VLAN IDs 101–150 and VLAN ID 1) and (S-VLAN IDs 201–250 and VLAN ID 2) and effectively replaces them with the new subrange encompassing S-VLAN IDs 101–250 and VLAN IDs 1–2.

```
host1(config-if)#vlan bulk-config test modify svlan-range 101 250 1 2
```

To separate the merged subranges, you can modify the unified subrange and add subranges as needed, provided that no dynamic VLAN subinterfaces currently exist for any VLAN ID within those subranges.

If you merge subranges by using SNMP, the new merged subrange takes the lowest instance value of the incorporated subranges. For example, if a VLAN range has three subranges with instance values of 2, 4, and 5 and the subranges with instance values of 2 and 5 are merged, the new merged subrange has an instance value of 2.

Changing the Administrative State of VLAN Subranges

VLAN subranges have an administrative state that enables you to remove dynamic VLAN subinterfaces on various subranges that belong to a single VLAN range. This functionality is important because subrange removal requires that no dynamic VLAN subinterfaces exist for any circuit on that subrange. The removal operation fails if any such interfaces exist.

By default, the administrative state of a VLAN subrange is up. When you change the administrative state to down by using the **vlan bulk-config shutdown** command, the router deletes all dynamic VLAN subinterfaces on the affected subranges. You can use the **show vlan subinterface** command to monitor the progress of the removal of all dynamic VLAN subinterfaces for the specified subrange.

No additional dynamic VLAN subinterfaces can be created for the subrange until you restore the administrative state to up by using the **no vlan bulk-config shutdown** command.

The following example specifies the original VLAN subranges.

```
host1(config-if)#vlan bulk-config test svlan-range 101 150 1 1
svlan-range 201 250 2 2 svlan-range 501 550 5 5 svlan-range 301 350 3 3
```

You cannot specify a partial subrange; the specified subrange must exactly match a subrange that has already been configured. The following command changes the administrative state of the second subrange (S-VLAN IDs 201–250 and VLAN ID 2) to down. The router removes all dynamic interface columns built on any of the VLAN IDs or S-VLAN IDs in this subrange. No additional dynamic VLAN subinterfaces can be created until you change the administrative state to up.

```
host1(config-if)#vlan bulk-config test shutdown svlan-range 201 250 2 2
```

The following command changes the administrative state of this same VLAN subrange to up.

```
host1(config-if)#no vlan bulk-config test shutdown svlan-range 201 250 2 2
```

You can also change the administrative state of VLAN subranges that are based on agent-circuit-identifier information. For example, assume that the following command is issued to configure a VLAN subrange based on agent-circuit-identifier information:

```
host1(config-if)#vlan bulk-config myNewBulkConfig svlan-range 50 100
agent-circuit-identifier
```

The following command changes the administrative state of this same VLAN subrange to down:

```
host1(config-if)#vlan bulk-config myNewBulkConfig shutdown svlan-range 50 100
agent-circuit-identifier
```

You can change the administrative state of all subranges in a bulk-configured VLAN range at the same time by issuing the command without specifying any subranges. When you shut down a named bulk configuration, all VLAN ranges belonging to that bulk configuration, including those based on double-tagged S-VLANs or agent-circuit-identifier information, are disabled.

The following command shuts down all four subranges belonging to the named VLAN range, test, regardless of their current state.

```
host1(config-if)#vlan bulk-config test shutdown
```

The time required for the router to complete an administrative state change depends on the number of VLAN subranges configured.

vlan bulk-config

- Use to create a bulk-configured VLAN range on a static VLAN major interface for use by a dynamic VLAN subinterface.
- For detailed information about how to use this command, see **vlan bulk-config** on page 589.
- Example

  ```
  host1(config)#vlan bulk-config test1 svlan-range 200 250 2
  ```

- Use the **no** version to remove the specified VLAN range from the VLAN major interface, to remove the specified subranges from the specified VLAN range, or to remove all subranges from the specified VLAN range. The **no** version also removes any overriding profile assignments for VLAN IDs or S-VLAN IDs within the deleted VLAN range or VLAN subrange.

vlan bulk-config modify

- Use to expand or shorten the range of the specified VLAN subrange. You can modify only a single specific subrange at a time.
- You can expand a subrange if none of the added VLAN IDs or S-VLAN IDs overlap with any other subrange. You can shorten a subrange if none of the dropped VLAN IDs or S-VLAN IDs have existing dynamic VLAN subinterfaces. You can also modify an existing subrange by configuring it to use agent-circuit-identifier information rather than a range of VLAN IDs.
- Modifying a subrange so that it completely includes at least one other subrange from within the same VLAN range effectively merges the subranges. Each subrange that is merged with another frees up a subrange for subsequent configuration. The subranges that are merged do not need to be adjacent to each other.
- The router retains any overriding profiles assigned to a subrange if the assignment falls within the modified subrange. If the assignment falls outside of the newly modified subrange, the router drops the overriding profile assignment. If two subranges are merged, the router retains overriding profiles that were assigned to the separate subranges and applies the overriding profiles to the newly merged subrange.

- Example

  host1(config-if)#**vlan bulk-config test modify svlan-range 200 250 1 3**

- There is no **no** version.

vlan bulk-config shutdown

- Use to administratively disable (shut down) a specified VLAN subrange or all subranges in a VLAN range. The administrative state of a VLAN subrange is enabled by default.
- Disabling the VLAN subrange deletes all dynamic VLAN subinterfaces on the affected subranges. You can use the **show vlan subinterface** command to monitor the progress of the removal of all dynamic VLAN subinterfaces for the specified subrange.
- No dynamic VLAN subinterfaces can subsequently be created for the subrange until you restore the administrative state to enabled by using the **no vlan bulk-config shutdown** command.
- Example

  host1(config-if)#**vlan bulk-config test shutdown svlan-range 200 250 1 3**

- Use the **no** version to enable the specified VLAN subrange or all subranges in a VLAN range.

Configuring Static VLAN Subinterfaces Within VLAN Subranges

You can do either of the following on an E-series router:

- Create a static VLAN subinterface within an existing bulk-configured VLAN subrange
- Create a bulk-configured VLAN subrange that includes an existing static VLAN subinterface

The following sections describe how to perform these tasks.

The example procedures in this section show how to configure static VLAN subinterfaces within VLAN subranges by using the same loopback interface referenced by multiple unnumbered IP interfaces. Instead of assigning a different IP address to each physical interface, the first example assigns an IP address to a loopback interface (loopback 0). Each physical interface is then configured as an unnumbered IP interface, referencing the same loopback interface.

Creating Static VLAN Subinterfaces Within VLAN Subranges

You can configure a static VLAN subinterface with a VLAN whose VLAN ID falls within an existing bulk-configured VLAN subrange.

To create a static VLAN subinterface within a VLAN subrange:

1. Create the VLAN major interface.

   ```
   host1(config)#interface gigabitEthernet 0/0
   host1(config-if)#encapsulation vlan
   ```

2. Create a bulk-configured VLAN range that includes one or more VLAN subranges.

   ```
   host1(config-if)#vlan bulk-config test vlan-range 200 250
   ```

3. Create a static VLAN subinterface by adding a subinterface number to the interface identification command.

   ```
   host1(config-if)#interface gigabitEthernet 0/0.2100
   ```

4. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     ```
     host1(config-if)#vlan id 201
     ```

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     ```
     host1(config-if)#vlan id 201 mac-address 0090.1a01.1234
     ```

5. To fully configure the VLAN subinterface, assign an IP address, or make it unnumbered.

   ```
   host1(config-if)#ip unnumbered loopback 0
   ```

Creating VLAN Subranges That Include Static VLAN Subinterfaces

You can configure a bulk-configured VLAN subrange that includes the VLAN ID belonging to an existing VLAN on a static VLAN subinterface. This example is essentially the reverse of the procedure in *Creating Static VLAN Subinterfaces Within VLAN Subranges* on page 599.

To create a VLAN subrange that includes a static VLAN subinterface:

1. Create the VLAN major interface.

   ```
   host1(config)#interface gigabitEthernet 3/1
   host1(config-if)#encapsulation vlan
   ```

2. Specify a static VLAN subinterface.

   ```
   host1(config-if)#interface gigabitEthernet 3/1.201
   ```

3. Do one of the following:

   - Assign a VLAN ID for the subinterface.

     host1(config-if)#**vlan id 201**

   - Assign a VLAN ID and the optional unique MAC address for the subinterface.

     host1(config-if)#**vlan id 201 mac-address 0090.1a01.1234**

4. Create a bulk-configured VLAN range that includes the VLAN ID of the previously configured VLAN. In this example, the VLAN range 100–250 includes VLAN ID 201.

   host1(config)#**interface gigabitEthernet 3/1**
   host1(config-if)#**vlan bulk-config test2 vlan-range 100 250**

5. To fully configure the VLAN subinterface, assign an IP address or make it unnumbered.

   host1(config-if)#**ip unnumbered loopback 0**

encapsulation vlan

- Use to configure VLAN as the encapsulation method on an interface.
- Issuing this command creates the VLAN major interface.
- Example

  host1(config-if)#**encapsulation vlan**

- Use the **no** version to disable VLAN encapsulation on the interface.

interface gigabitEthernet
interface tenGigabitEthernet

- Use to select a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.
- For information about specifying a Gigabit Ethernet or 10-Gigabit Ethernet interface, see **interface gigabitEthernet** and **interface tenGigabitEthernet** on page 586.
- Examples

  host1(config)#**interface gigabitEthernet 1/0**
  host1(config)#**interface gigabitEthernet 4/0/1**
  host1(config)#**interface tenGigabitEthernet 4/0/1**

- Use the **no** version to remove IP from an interface.

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.
- Examples

  host1(config-if)#**ip unnumbered fastEthernet 3/0**
  host1(config-if)#**ip unnumbered loopback 10**

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

vlan bulk-config

- Use to create a bulk-configured VLAN range on a static VLAN major interface for use by a dynamic VLAN subinterface.
- For detailed information about how to use this command, see **vlan bulk-config** on page 589.
- Example

  host1(config)#**vlan bulk-config test1 svlan-range 200 250 2 2**

- Use the **no** version to remove the specified VLAN range from the VLAN major interface, to remove the specified subranges from the specified VLAN range, or to remove all subranges from the specified VLAN range. The **no** version also removes any overriding profile assignments for VLAN IDs or S-VLAN IDs within the deleted VLAN range or VLAN subrange.

Monitoring Dynamic Interfaces and Profiles

You can use the **show** commands described in this section to monitor configurations created with dynamic interfaces and profiles.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

show atm aal5 interface

- Use to display information about a configured ATM AAL5 interface.
- Field descriptions
  - AAL5 Interface operational status—Operational status of the AAL5 interface: up, down, lowerLayerDown
  - time since last status change—Time since last reported change to the AAL5 interface operational status in hh:mm:ss format
  - SNMP trap link-status—Whether SNMP link status traps are enabled or disabled on the ATM AAL5 interface
  - Auto configure ATM 1483 status—Whether the autoconfiguration feature for a dynamic ATM 1483 subinterface configured over the ATM AAL5 interface is enabled or disabled

- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of incoming errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- OutDiscards—Number of outgoing packets discarded on this interface

- Example

```
host1#show atm aal5 interface atm 3/0
AAL5 Interface ATM 3/0 operational status:    lowerLayerDown
      time since last status change: 00:08:46

SNMP trap link-status: disabled
Auto configure ATM 1483 status: disabled

InPackets:        0
InBytes:          0
OutPackets:       0
OutBytes:         0
InErrors:         0
OutErrors:        0
InPacketDiscards: 0
OutDiscards:      0
```

show atm bulk-config

- Use to display information, including base profile assignments and overriding profile assignments, for the bulk-configured VC ranges on an ATM AAL5 interface.
- To display information for all VC ranges on the router, use the command with no keywords.
- To display information for all VC ranges on a specified ATM AAL5 interface, use the command with the **atm** keyword and interface specifier.
- To display information for the VC range associated with a particular bulk configuration name, use the command with the **name** keyword.
- To display information for a particular VC range on a specified ATM AAL5 interface, use the command with the **atm** keyword and interface specifier and the **name** keyword.
- To display information only about overriding profile assignments configured for specific ATM PVCs within bulk-configured VC subranges, use the command with the **override** keyword. When you specify the **override** keyword, the command does not display information about base profile assignments.

- Field descriptions
  - Interface—Identifier of the ATM AAL5 physical interface on which the bulk-configured VC range resides. For more information about specifying the ATM interface, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - Bulk Config Name—Name of the bulk-configured VC range on this interface
  - Start VPI—Starting virtual path identifier (inclusive) of the VC subrange
  - End VPI—Ending virtual path identifier (inclusive) of the VC subrange
  - Start VCI—Starting virtual circuit identifier (inclusive) of the VC subrange
  - End VCI—Ending virtual circuit identifier (inclusive) of the VC subrange
  - Assigned Profile—Base profile name for the dynamic ATM 1483 subinterface assigned to this VC subrange with the **profile atm1483 bulk-config-name** command. When no profile is assigned to the VC subrange, the field displays none assigned.
  - Admin State—Administrative state of the VC subrange: up or down
  - Profile override(s)—When overriding profile assignments are configured on the router, the command displays the following fields:
    - Interface—Identifier of the ATM AAL5 physical interface
    - Bulk Config Name—Name of the bulk-configured VC range on this interface that includes the VC subrange encompassing the specified ATM PVC
    - VPI—Virtual path identifier of the PVC within the bulk-configured VC subrange
    - VCI—Virtual circuit identifier of the PVC within the bulk-configured VC subrange
    - Assigned Profile—Name of the overriding profile assigned to the specified PVC with the **profile atm1483 bulk-config-name pvc** command
    - Status—Operational status of the overriding profile assignment: Active or Inactive. Active indicates that the router uses the overriding profile to create dynamic interface columns because no static ATM circuits with the same VPI/VCI values exist on this interface. Inactive indicates that the router does not use the overriding profile to create dynamic interface columns because a static ATM circuit with the same VPI/VCI values exists on this interface.

- Example 1—Displays information about base profile assignments and overriding profile assignments for all bulk-configured VC ranges on the router. The VC range named test consists of a single VC subrange (1, 1, 101, 200), has a base profile named atm1483BaseProfile assigned, and has an overriding profile named overrideProfile1 assigned to two ATM PVCs within the VC subrange. The VC range named test2 is a placeholder range that has no VC subranges configured and no base profile assigned.

```
host1#show atm bulk-config

             Bulk
            Config Start End Start End                    Admin
 Interface   Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0  test     1   1   101 200 atm1483BaseProfile    up
ATM AAL5 3/2  test2    --  --   --- ---      none assigned   ---

2 bulk configuration(s) found

Profile override(s):

             Bulk
            Config
 Interface   Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0  test   1 151 overrideProfile1 Active
ATM AAL5 3/0  test   1 161 overrideProfile1 Active
2 profile override(s) found
```

- Example 2—Displays information about base profile assignments and overriding profile assignments for all VC ranges configured on a specified ATM AAL5 interface

```
host1#show atm bulk-config atm 3/0

             Bulk
            Config Start End Start End                    Admin
 Interface   Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0  test     1   1   101 200 atm1483BaseProfile    up
1 bulk configuration(s) found

Profile override(s):

             Bulk
            Config
 Interface   Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0  test   1 151 overrideProfile1 Active
ATM AAL5 3/0  test   1 161 overrideProfile1 Active
2 profile override(s) found
```

- Example 3—Displays information about base profile assignments and overriding profile assignments for a particular bulk-configured VC range

```
host1#show atm bulk-config name test

              Bulk
             Config Start End Start End                    Admin
 Interface    Name   VPI  VPI  VCI  VCI  Assigned Profile  State
------------ ------ ----- --- ----- --- ------------------ -----
ATM AAL5 3/0   test     1   1   101 200 atm1483BaseProfile    up
1 bulk configuration(s) found

Profile override(s):

              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

- Example 4—Displays information only about overriding profile assignments for all bulk-configured VC ranges on the router

```
host1#show atm bulk-config override

Profile override(s):

              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

- Example 5—Displays information only about overriding profile assignments for a particular VC range configured on a specified ATM AAL5 interface

```
host1#show atm bulk-config atm 3/0 override

Profile override(s):

              Bulk
             Config
 Interface    Name  VPI VCI Assigned Profile Status
------------ ------ --- --- ---------------- ------
ATM AAL5 3/0   test   1 151 overrideProfile1 Active
ATM AAL5 3/0   test   1 161 overrideProfile1 Active
2 profile override(s) found
```

show atm subinterface

- Use to display the current state of all ATM subinterfaces, all ATM subinterfaces configured on a specified ATM physical interface, or a specific ATM subinterface.
- To specify an ATM subinterface for ERX-7xx models, ERX-14xx models, and ERX-310 routers, use the *slot*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *port*—Port number on the I/O module
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To specify an ATM subinterface for E120 and E320 routers, use the *slot*/*adapter*/*port*.*subinterface* format.
  - *slot*—Number of the chassis slot
  - *adapter*—Identifier for the IOA within the E320 chassis, either 0 or 1, where:
    - 0 indicates that the IOA is installed in the right IOA bay (E120 router) or the upper IOA bay (E320 router).
    - 1 indicates that the IOA is installed in the left IOA bay (E120 router) or the lower IOA bay (E320 router).
  - *port*—Port number on the IOA
  - *subinterface*—Number of the subinterface in the range 1–2147483647
- To display brief summary information for all ATM subinterfaces, or for ATM subinterfaces configured on a specified ATM physical interface, use the **summary** keyword.
- To display status information only for ATM subinterfaces with a specific operating status, use the **status** keyword with one of the following status values. (See the Status field description for an explanation of these values.)
  - dormant
  - dormantLockout
  - down
  - lowerLayerDown
  - notPresent
  - up

- To display the current state of an ATM subinterface created on the PVC with the specified VPI and VCI values, use the **atm** *slot/port/vpi/vci* format (for ERX-7xx models, ERX-14xx models, and ERX-310 routers) or the *slot/adapter/port/vpi/vci* format (for E120 and E320 routers) to identify the ATM subinterface (Example 5).

NOTE: You can use the **atm** *slot/port/vpi/vci* format as an alternative to the **atm** *slot/port.subinterface* format with the specific **show interface** and **show subinterface** commands to monitor all ATM 1483 subinterfaces (except NBMA interfaces) as well as the upper-layer interfaces configured over an ATM 1483 subinterface. You cannot, however, use the **atm** *slot/port/vpi/vci* format to create or modify an ATM 1483 subinterface.

These guidelines also apply to E120 and E320 routers when you use the **atm** *slot/adapter/port/vpi/vci* format as an alternative to the **atm** *slot/adapter/port.subinterface* format.

- For more information, see *Creating a Basic Configuration* in *Chapter 1, Configuring ATM*.
- Field descriptions
  - Interface—Interface identifier
  - ATM-Prot—One of the following ATM protocol types:
    - RFC-1483—Multiprotocol encapsulation over AAL5
    - NBMA—Nonbroadcast multiaccess interface
    - ATM/MPLS—Local ATM passthrough interface
  - VCD—Virtual circuit descriptor
  - VPI—Virtual path identifier
  - VCI—Virtual circuit (or channel) identifier
  - Circuit Type—Type of circuit: PVC
  - Encap—Administered encapsulation method based on what was configured with the **atm pvc** command
  - MTU—Maximum transmission unit size for the interface
  - Status—One of the following ATM 1483 subinterface states:
    - absent—Represents the notPresent state and indicates that, although the SRP detects the ATM 1483 subinterface, the module on which the subinterface resides has not completed booting up, has failed, or is disabled.
    - dormant—Indicates that the ATM 1483 subinterface is performing autodetection of one or more upper-layer encapsulation types and is waiting to receive a packet of that type on a lower-layer interface. An ATM 1483 subinterface transitions from the dormant state to the up state when the router receives a valid packet of the specified encapsulation type on the interface.

  - dormantLockout—Indicates that a dormant ATM 1483 subinterface has one or more upper-layer encapsulation types currently undergoing encapsulation type lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the dormant state when the lockout time expires for all upper-layer encapsulation types undergoing lockout. An ATM 1483 subinterface transitions from the dormantLockout state to the up state when the router receives a valid packet for an encapsulation type that is configured for autodetection but is not undergoing lockout.
  - down—Indicates that the ATM 1483 subinterface is administratively disabled or has a circuit that is down or not configured.
  - lowerLayerDown—Indicates that a lower-layer interface below the ATM 1483 subinterface is down.
  - up—Indicates that the ATM 1483 subinterface is up and able to transfer data. For an ATM 1483 subinterface that supports one or more dynamic upper-layer interfaces, indicates that the router has created the dynamic upper-layer interface or is in the process of creating it.
- Interface Type—Type of ATM 1483 subinterface: dynamic or static
- Auto configure status—Setting of the autoconfiguration feature
  - dynamic—Autodetection is on; the router automatically detects the next upper interface
  - static—Autodetection is off
- Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: bridged Ethernet, IP, PPP, or PPPoE
- Detected 1483 encapsulation—If the encapsulation type is set to **aal5autoconfig**, displays the 1483 encapsulation type detected on the subinterface (displays AUTO until a packet is detected)
- Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: bridged Ethernet, IP, PPP, PPPoE, or (if no packet has been received) none
- Interface types in lockout—Encapsulation types currently experiencing lockout: bridged Ethernet, IP, PPP, PPPoE, or none
- Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
  - Min—Minimum lockout time, in seconds
  - Max—Maximum lockout time, in seconds
  - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
  - Next—Lockout time for the router to use for the next lockout event, in seconds
- Assigned profile—For each dynamic interface type, indicates whether or not a profile is assigned and, if assigned, displays the profile name

- Subscriber info—Subscriber login information for the specified dynamic interface type (bridged Ethernet or IP)
- SNMP trap link-status—Trap link status: enabled or disabled
- InPackets—Number of packets received on this interface
- InBytes—Number of bytes received on this interface
- OutPackets—Number of packets transmitted on this interface
- OutBytes—Number of bytes transmitted on this interface
- InErrors—Number of errors received on this interface
- OutErrors—Number of outgoing errors on this interface
- InPacketDiscards—Number of incoming packets discarded on this interface
- InPacketsUnknownProtocol—Number of incoming packets with an unknown protocol type
- OutDiscards—Number of outgoing packets discarded on this interface

- Example 1—Displays the current state of all ATM subinterfaces

```
host1#show atm subinterface
                                    Circuit                           Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
ATM 2/0.102 RFC-1483 102   0 102 PVC     AUTO  9180 up             Dynamic
ATM 2/0.103 RFC-1483 103   0 103 PVC     AUTO  9180 dormant        Static
3 interface(s) found
```

- Example 2—Displays summary information for all ATM subinterfaces shown in Example 1

```
host1#show atm subinterface summary
3 subinterfaces: 1 up, 0 down,
1 dormant, 1 dormantLockout,
0 lowerLayerDown, 0 not present
```

- Example 3—Displays status information for all ATM subinterfaces in the dormantLockout state

```
host1#show atm subinterface status dormantLockout
                                    Circuit                           Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static
1 interface(s) found
```

- Example 4—Displays the current state of a specific ATM subinterface

```
host1#show atm subinterface atm 2/0.101
                                    Circuit                           Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU      Status         Type
----------- -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 dormantLockout Static

Auto configure status         : dynamic
Auto configure interface(s)   : IP  bridgedEthernet  PPP  PPPoE
Detected 1483 encapsulation   : AUTO
Detected dynamic interface    : none
Interface types in lockout    : IP
```

```
Lockout state (seconds)       : Min Max  Current Elapsed Next
------------------------------ --- ---- ------- ------- ----
IP                              1   30      16       7   30
BridgedEnet                   900 3600       0       0  900
PPP                             1  300       0       0    1
PPPoE                           1  300       0       0    1

Assigned profile (IP)         : ipoa
Assigned profile (BridgedEnet): beth
Assigned profile (PPP)        : ppptest
Assigned profile (PPPoE)      : pppoetest
Assigned profile (any)        : none assigned

BridgedEnet subscriber info   :
Username: elaine@jpeterman.com
Password: putty
Authenticate: enabled

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  1904
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         14
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

- Example 5—Displays the current state of a specific ATM subinterface created on the PVC with the specified VPI and VCI values

```
host1#show atm subinterface atm 0/0/0/101
                                  Circuit                    Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     AUTO  9180 up     Static

Auto configure status         : dynamic
Auto configure interface(s)   : PPPoE
Detected 1483 encapsulation   : SNAP
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
PPPoE                           1 300       0       0    1

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : pppoeprofile
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled

InPackets:                5119
InBytes:                  358672
OutPackets:               5107
OutBytes:                 357510
InErrors:                 0
```

```
OutErrors:                0
InPacketDiscards:         3
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

show atm vc

- Use to display a summary of all configured ATM VCs and reserved VC ranges.
- You can specify one or more of the following keywords individually or in combination:
  - **vpi**—Displays VCs on a specific VPI
  - **category**—Displays VCs that have a specific service category
  - **status**—Displays VCs with a certain status
- To display only a summary of all reserved VC ranges on the router, specify the **reserved** keyword with no other keywords. This includes VPI/VCI ranges reserved for use by dynamic ATM 1483 subinterfaces.
- Field descriptions
  - Interface—Interface identifier
  - VPI—Virtual path identifier
  - VCI—Virtual channel identifier
  - VCD—Virtual circuit descriptor
  - Type—Type of circuit: PVC
  - Encap—Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
  - Category—Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, or CBR
  - Rx/Tx Peak—Peak rate, in Kbps
  - Rx/Tx Avg—Average rate, in Kbps
  - Rx/Tx Burst—Maximum number of cells that can be burst at the peak cell rate
  - Status—State of the virtual circuit: Up or Down
  - Start VPI—Starting virtual path identifier (inclusive) of the reserved VC range
  - Start VCI—Starting virtual circuit identifier (inclusive) of the reserved VC range
  - End VPI—Ending virtual path identifier (inclusive) of the reserved VC range
  - End VCI—Ending virtual circuit identifier (inclusive) of the reserved VC range

- Example 1—Displays all VCs and reserved VC ranges on the router

```
host1#show atm vc
                                          Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface    VPI VCI  VCD  Type Encap gory Peak   Avg  Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found


Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

- Example 2—Displays all reserved VC ranges on the router

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

show columns

- Use to display static and dynamic interface counts for each interface column.
- Counts for PPP and PPPoE interface columns are updated when the PPP layer comes up.
- Counts for bridged Ethernet and IP over ATM columns are updated when the ATM layer comes up.
- Field descriptions
  - Type—Interface type
  - Total—Total number of interfaces on this column
  - Static—Number of static interfaces on this column
  - Dynamic—Number of dynamic interfaces on this column

- Example

```
host#show columns
                  Interface columns:
                  ------------------
        Type           Total      Static    Dynamic
-------------------- ---------- ---------- ----------
Bridged Ethernet              4          2          2
IP over ATM                   4          2          2
PPP                          22         12         10
PPPoE                        10          5          5
```

show pppoe interface

- Use to display summary information about the encapsulation type lockout parameters configured for PPPoE clients on a dynamic PPPoE subinterface column.
- The following field descriptions and example include only the portion of the **show pppoe interface** command display relevant to lockout configuration for PPPoE clients. For more information about using this command, see **show pppoe interface** in *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*.
- Field descriptions
  - Lockout Configuration (seconds)—Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column
    - Min—Minimum lockout time, in seconds
    - Max—Maximum lockout time, in seconds
    - Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout
    - Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see *Guidelines for Configuring Encapsulation Type Lockout* in *Chapter 15, Configuring Dynamic Interfaces*.
- Example

```
host1#show pppoe interface atm 3/0.101
. . .
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockouts: 0
 Total clients in lockout grace period: 0
```

show pppoe interface lockout-time

- Use to display detailed information about the current encapsulation type lockout condition for each PPPoE client associated with a dynamic PPPoE subinterface column on a static PPPoE major interface.
- Field descriptions
  - PPPoE interface—Specifier for the PPPoE interface
  - Lockout Configuration (seconds)—Encapsulation type lockout settings for the PPPoE client associated with the dynamic PPPoE subinterface column
    - Min—Minimum lockout time, in seconds
    - Max—Maximum lockout time, in seconds
    - Total clients in active lockouts—Number of PPPoE clients currently undergoing dynamic encapsulation type lockout
    - Total clients in lockout grace period—Number of PPPoE clients currently in a lockout grace period; for more information about the lockout grace period, see *Guidelines for Configuring Encapsulation Type Lockout* in *Chapter 15, Configuring Dynamic Interfaces*
  - Client Address—Source MAC address of the PPPoE client
  - Current—Current lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
  - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if the PPPoE client is not undergoing lockout
  - Next—Lockout time that the router uses for the next lockout event, in seconds
- Example

```
host1#show pppoe interface atm 3/0.101 lockout-time
PPPoE interface ATM 3/0.101
Lockout Configuration (seconds): Min 5, Max 60
 Total clients in active lockout: 0
 Total clients in lockout grace period: 0
Client Address Current Elapsed Next
-------------- ------- ------- ----
0090.1a10.165e       0       0    5
```

show pppoe subinterface

- Use to display the source MAC address of a PPPoE client when a subscriber is connected to the router through an available PPPoE session. You can then specify this MAC address in the **pppoe clear lockout interface** command to clear the lockout condition for the PPPoE client.
- To display configuration, status, and statistics information, including the source MAC address of the PPPoE client, use the **full** keyword.
- The following field descriptions and example include only the portion of the **show pppoe subinterface** command display relevant to the source MAC address for PPPoE clients. For more information about using this command, see **show pppoe subinterface** in *Chapter 10, Configuring Point-to-Point Protocol over Ethernet*.

- Field descriptions
  - PPPoE subinterface—Specifier for the PPPoE subinterface
  - source MAC address—MAC address of the PPPoE client associated with the dynamic PPPoE subinterface column
- Example

```
host1#show pppoe subinterface full
...
    PPPoE subinterface ATM 3/0.101 has source MAC address 0090.1a10.165e
...
```

show profile

- Use to display information about profiles.
- To display information about a specific profile, use the **name** keyword.
- To display a list of profiles configured on the router, use the **brief** keyword.
- Field descriptions
  - Profile—Name of the profile that is displayed
  - 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—Name of the virtual router (VR) assigned to the profile; interfaces created by the profile are attached to this VR
  - Directed Broadcast—Enabled or disabled
  - ICMP Redirects—Enabled or disabled
  - Access Route Addition—Enabled or disabled
  - Network Address Translation—Enabled or disabled; domain location (inside or outside)
  - Source-Address Validation—Enabled or disabled
  - Ignore DF Bit—Enabled or disabled
  - Filter Option Packets—Router filters out packets with IP options; enabled or disabled
  - Administrative MTU—MTU size configured on the profile
  - TCP MSS value—Maximum segment size for TCP SYN packets traveling through the interface
  - Inactivity Timer—Inactivity timer setting; enabled or disabled
  - Route Map Name—Route map applied to the IP interface subscriber; enabled or disabled
  - 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

- IGMP—Enabled or disabled
- static-groups—Displays address of any static groups configured for IGMP
- Input policy—Name of input policy and whether statistics are enabled or disabled
- Output policy—Name of output policy and whether statistics are enabled or disabled
- PPP Keepalive—PPP keepalive period, in seconds
- PPP Magic Number—Enabled or disabled
- PPP Peer DNS Priority—Enabled or disabled
- PPP Peer WINS Priority—Enabled or disabled
- PPP Authentication—Type of authentication configured: PAP, CHAP, or none
- PPP Authentication Router—Name of authentication virtual router
- PPP Negotiate MRU—MRU configured for the profile
- PPP Packet Log—Enabled or disabled
- PPP State Log—Enabled or disabled
- PPP Chap Challenge Length—Minimum and maximum Chap Challenge length
- PPP Passive Mode—Enabled or disabled
- PPP Multilink—Enabled or disabled
- PPP IPCP netmask option—Enabled or disabled
- PPP AAA Profile—AAA profile associated with this PPP interface
- PPP Multilink Fragmentation—Enabled or disabled
- PPP Multilink Fragment Size—Multilink fragment size for this PPP interface
- PPP Multilink Reassembly—Enabled or disabled
- PPP Multilink Mrru—Multilink MRRU value for this PPP interface
- PPP Initiate IP—Initiation of IPv4 over this PPP interface; enabled or disabled
- PPP Initiate IPv6—Initiation of IPv6 over this PPP interface; enabled or disabled
- PPPoE Max Sessions—Maximum number of PPPoE subinterfaces that can be on an interface
- PPPoE Always-offer—Router offers to set up a session for the client, even if the router has insufficient resources to establish a session; enabled or disabled
- PPPoE Remote-Circuit-Id—Router captures and processes a vendor-specific tag containing a remote circuit ID transmitted from a digital subscriber line access multiplexer (DSLAM); enabled or disabled
- PPPoE Log PPPoeControlPacket—Enabled or disabled
- PPPoE duplicate-protect—Enabled or disabled

- PPPoE ACNAME—Access concentrator name
- PPPoE URL—URL sent in PADM message to PPPoE clients
- PPPoE MOTM—Message of the minute sent in the PADM message to PPPoE clients
- PPPoE Service-Name Table—Name of the PPPoE service name table, if configured for the specified profile
- ATM1483 Auto-configure—Whether autodetection of the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) is enabled or disabled for a dynamic ATM 1483 subinterface
- ATM1483 lockout (seconds)—Encapsulation type lockout setting for the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE) configured on a dynamic ATM 1483 subinterface
  - range—Minimum lockout time–maximum lockout time, in seconds
  - no lockout—Encapsulation type lockout is disabled
- ATM1483 PVC circuit type—Encapsulation setting for the PVC configured on a dynamic ATM 1483 subinterface
  - aal5autoconfig—Enables autodetection of the 1483 encapsulation (LLC/SNAP or VC multiplexed)
  - aal5mux ip—VC-based multiplexed circuit for IP only
  - aal5snap—LLC encapsulated circuit; the LLC/SNAP header precedes the protocol datagram
- ATM1483 PVC service category—Service type setting for the PVC configured on a dynamic ATM 1483 subinterface: UBR (the default), UBR PCR, NRT-VBR, RT-VBR, or CBR
- ATM1483 PVC Peak rate—Peak cell rate (PCR), in Kbps, for the PVC configured on a dynamic ATM 1483 subinterfaces
- ATM1483 PVC Avg rate—Average rate, in Kbps, for the PVC configured on a dynamic ATM 1483 subinterface; also referred to as sustained cell rate (SCR)
- ATM1483 PVC Burst size—Length in cells of the burst for the PVC configured on a a dynamic ATM 1483 subinterface; also referred to as maximum burst size (MBS)
- ATM1483 Description—Text description assigned to ATM 1483 subinterfaces that are created with this profile
- ATM1483 Advisory Rx Speed—Configured receive speed, in Kbps, for the dynamic ATM 1483 subinterface. The E-series LAC sends this value to the LNS in the RX Connect-Speed AVP [38].
- ATM1483 PVC OAM Administrative status—Status of OAM F5 loopback cell generation (for VC integrity) on a circuit created with this profile: enabled or disabled
- ATM1483 PVC OAM Loopback frequency—Number of seconds between transmissions of OAM F5 end-to-end loopback cells on a circuit created with this profile

- ATM1483 Ip Subscriber information—Subscriber login information for the specified dynamic interface type
- ATM1483 Profile—Name of the profile assigned to the specified upper-interface encapsulation type (bridged Ethernet, IP, PPP, or PPPoE); these profiles are referenced in the base profile for a dynamic ATM 1483 subinterface as nested profile assignments
- ATM Virtual Circuit Class—Name of the ATM VC class assigned to the bulk-configured VC ranges associated with this base profile, if configured
- VLAN Auto-configure—Whether autodetection of the specified upper-interface encapsulation type (IP or PPPoE) is enabled or disabled for a dynamic VLAN subinterface
- VLAN Agent Circuit Identifier— Whether autodetection of the VLAN subinterface uses the agent-circuit-identifier information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets: enabled or disabled
- VLAN Advisory Rx Speed—Configured advisory receive speed, in Kbps, for the dynamic VLAN subinterface; the E-series LAC sends this value to the LNS in the RX Connect-Speed AVP [38]
- VLAN Advisory Tx Speed—Configured advisory speed, in Kbps, for the dynamic VLAN subinterface.
- VLAN Description—Text description assigned to VLAN subinterfaces that are created with this profile
- VLAN Profile—Name of the profile assigned to the specified upper-interface encapsulation type (IP or PPPoE); these profiles are referenced in the base profile for a dynamic VLAN subinterface as nested profile assignments
- VLAN Service Profile—Service profile name for a VLAN
- VLAN Svlan Ethertype—Ethertype that the packet must use this to create the dynamic VLAN subinterface
- Bridged Ethernet Mtu—MTU size configured for a dynamic bridged Ethernet interface
- Bridged Ethernet Service Profile—Name of the IP service profile associated with the interface profile for this dynamic bridged Ethernet interface

- Example 1—Displays configuration information for a profile assigned to a dynamic interface

```
host1#show profile name pppoeProfile
Profile                    : pppoeProfile
Unnumbered interface on    : loopback 1
Router                     : default
Directed Broadcast         : Disabled
ICMP Redirects             : Disabled
Access Route Addition      : Enabled
Network Address Translation: Disabled
Source-Address Validation  : Disabled
Ignore DF Bit              : Disabled
Filter Option Packets      : Disabled
Administrative MTU         : 1500
TCP MSS value              : 0
Inactivity Timer           : Disabled
Route Map Name             : Disabled
Auto Detect                : Disabled
Auto Configure             : Disabled

IGMP                       : Enabled
  static-groups            :
  Input policy:  bobb statistics enabled
  Output policy: bobb statistics enabled

PPP Keepalive               : 30
PPP Magic Number            : enabled
PPP Peer DNS Priority       : disabled
PPP Peer WINS Priority      : disabled
PPP Authentication          : pap/chap
PPP Authentication Router   :
PPP Negotiate MRU           : (use lower layer MRU)
PPP Packet Log              : disabled
PPP State Log               : disabled
PPP Chap Challenge Length   : 16 - 32
PPP Passive Mode            : disabled
PPP Multilink               : disabled
PPP IPCP Netmask Option     : disabled
PPP AAA Profile             :
PPP Multilink Fragmentation : disabled
PPP Multilink Fragment Size : (use MTU)
PPP Multilink Reassembly    : disabled
PPP Multilink Mrru          : (use MRU)
PPP Initiate IP             : disabled
PPP Initiate IPv6           : disabled
PPPoE Max Sessions          : 2
PPPoE Always-offer          : Disabled
PPPoE Remote-Circuit-Id     : Enabled
PPPoE Log PPPoeControlPacket: Disabled
PPPoE duplicate-protect     : Enabled
PPPoE ACNAME                : CYM9876
PPPoE URL                   : http://www.urlofinterest.com
PPPoE MOTM                  : goodmorning
PPPoE Service-Name table    : myServiceTable1
```

- Example 2—Displays configuration information for a base profile assigned to a dynamic ATM 1483 subinterface

```
host1#show profile name atm1483BaseProfile
ATM1483 Auto-configure ip                 : disabled
ATM1483 Auto-configure bridgedEthernet    : disabled
ATM1483 Auto-configure ppp                : enabled
ATM1483 lockout (seconds) ppp             : range : 1-300
ATM1483 Auto-configure pppoe              : enabled
ATM1483 lockout (seconds) pppoe           : range : 1-300
ATM1483 PVC circuit type         :  aal5autoconfig
ATM1483 PVC service category     :  Nrt-Vbr
ATM1483 PVC Peak rate : 10000, Avg rate : 2000, Burst size : 500
ATM1483 Description              :  VC_atm1483
ATM1483 Advisory Rx Speed        :  2000000000

ATM1483 PVC OAM Administrative status: enabled
ATM1483 PVC OAM Loopback frequency:    30

ATM1483 Ip Subscriber information:
  user             : elaine
  domain           : jpeterman.com
  password         : putty
ATM1483 IP Profile               : none assigned
ATM1483 Bridged Ethernet Profile : none assigned
ATM1483 PPP Profile              : none assigned
ATM1483 PPPoE Profile            : pppoeProfile
ATM Virtual Circuit Class        : premium-subscriber-class
```

- Example 3—Displays configuration information for a base profile assigned to a dynamic VLAN subinterface

```
host1#show profile name vlanProfile
VLAN Auto-configure ip             : enabled
VLAN Auto-configure pppoe          : enabled
VLAN Svlan Ethertype               : auto-configure
VLAN Agent Circuit Identifier      : disabled
VLAN Advisory Rx Speed             : 100 Kbps
VLAN Advisory Tx Speed             : 2500 Kbps
VLAN Description                   : testing
VLAN IP Profile                    : ipProfile
VLAN PPPoE Profile                 : pppoeProfile
VLAN Service Profile               : none assigned
Bridged Ethernet Mtu               : 1971
Bridged Ethernet Service Profile   : eastServiceProfile
```

show vlan bulk-config

- Use to display information, including base profile assignments and overriding profile assignments, for the bulk-configured VLAN ranges on a VLAN major interface.
- To display information for all VLAN ranges on the router, use the command with no keywords.
- To display information for the VLAN range associated with a particular bulk configuration name, use the command with the **name** keyword.
- To display information for a particular VLAN range on a specified VLAN interface, use the command with the interface specifier and the **name** keyword.

- Field descriptions
  - Interface—Identifier of the physical interface on which the bulk-configured VLAN range resides. For more information about specifying the VLAN subinterface, see *Interface Types and Specifiers* in *JUNOSe Command Reference Guide, About This Guide*.
  - Bulk Config Name—Name of the bulk-configured VLAN range on this interface
  - Start Svlan Id—Starting S-VLAN ID (inclusive) of the S-VLAN group in the subrange
  - End Svlan Id—Ending S-VLAN ID (inclusive) of the S-VLAN group in the subrange
  - Start Vlan Id—Starting VLAN ID (inclusive) of the VLAN group in the subrange
  - End Vlan Id—Ending VLAN ID (inclusive) of the VLAN group in the subrange
  - Assigned Profile—Base profile name for the dynamic VLAN subinterface assigned to this VLAN subrange with the **profile vlan bulk-config** command. When no profile is assigned to the VLAN subrange, the field displays none assigned.
  - Admin State—Administrative state of the VLAN subrange: up or down
- Example 1—Displays information about base profile assignments and overriding profile assignments for all bulk-configured VLAN ranges on the router

```
host1#show vlan bulk-config
                  Bulk   Start  End  Start End
                 Config  Svlan Svlan Vlan  Vlan   Assigned
   Interface      Name    Id    Id    Id    Id     Profile    Status
---------------- -------- ----- ----- ----- ---- ------------- ------
FastEthernet 4/6 vlanOnly     1     1     0    0 vlanProfile   Up
FastEthernet 4/6 vlanOnly     2     2   any  any vlanProfile   Up
FastEthernet 0/5 vlanOnly ----- ----- ----- ---- none assigned ------
FastEthernet 4/0 vlanOnly     2     2   any  any none assigned Up
% 4 vlan bulk-config(s) found

Profile override(s):

                  Bulk
                 Config  Svlan Vlan Assigned
   Interface      Name    Id    Id   Profile  Status
---------------- -------- ----- ---- --------- ------
FastEthernet 4/6 vlanOnly     2    3 ipProfile Active
FastEthernet 4/6 vlanOnly     2    4 ipProfile Active
% 2 profile override(s) found
```

- Example 2—Displays information about base profile assignments and overriding profile assignments for all VLAN ranges configured on a specified Fast Ethernet interface

```
host1#show vlan bulk-config interface fastEthernet 4/6
                  Bulk   Start  End  Start End
                 Config  Svlan Svlan Vlan  Vlan   Assigned
   Interface      Name    Id    Id    Id    Id     Profile    Status
---------------- -------- ----- ----- ----- ---- ------------- ------
FastEthernet 4/6 vlanOnly     1     1     0    0 vlanProfile   Up
FastEthernet 4/6 vlanOnly     2     2   any  any vlanProfile   Up

% 2 vlan bulk-config(s) found

Profile override(s):

                  Bulk
                 Config  Svlan Vlan Assigned
   Interface      Name    Id    Id   Profile  Status
---------------- -------- ----- ---- --------- ------
FastEthernet 4/6 vlanOnly     2    3 ipProfile Active
FastEthernet 4/6 vlanOnly     2    4 ipProfile Active
% 2 profile override(s) found
```

show vlan profile

- Use to display information about the dynamic VLAN subinterfaces that have been created with an overriding profile assignment.
- Use the **bulk-config** keyword to display information about bulk-configured ranges.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID for the interface
  - Assigned Profile—Overriding profile to be assigned to the VLAN
  - Status—Operational status of the overriding profile assignment: Active or Inactive. Active indicates that the router uses the overriding profile to create dynamic interface columns because no static VLAN subinterfaces exist on this interface. Inactive indicates that the router does not use the overriding profile to create dynamic interface columns because a static VLAN subinterface exists on this interface.
- Example

```
host1#show vlan profile override

Profile override(s):

                  Bulk
                 Config Svlan Vlan Assigned
   Interface      Name   Id    Id   Profile  Status
---------------- ------ ----- ---- --------- ------
FastEthernet 4/6 vlanB2  ----    2 ipProfile Active

% 1 profile override(s) found
```

show vlan subinterface

- Use to display configuration and status information for a specified VLAN subinterface or for all VLAN subinterfaces configured on the router.
- Use the **summary** keyword to display only the counts of all VLAN subinterfaces and VLAN major interfaces configured on the router.
- Use the **vlan** or **svlan** keywords to display information about specific VLAN IDs or S-VLAN IDs.
- Use the **agent-circuit-identifier** keyword to display information about VLAN subinterfaces that are created based on the agent-circuit-id information in the option 82 field of DHCP messages or in the DSL Forum VSA 26-1 of PPPoE PADR and PADI packets. Using this keyword causes the router to display the agent-circuit-identifier string in the command output.
- Field descriptions
  - Interface—Type and specifier of the VLAN subinterface
  - Status—Status of the VLAN subinterface: up, down, dormant, lowerLayerDown, absent
  - MTU—Maximum allowable size (in bytes) of the MTU for the VLAN subinterface
  - Svlan Id—S-VLAN ID value, if configured
  - Vlan Id—VLAN ID value for the VLAN subinterface
  - Ethertype—S-VLAN Ethertype value, if configured
  - Type—Type of VLAN subinterface
    - Static—VLAN or S-VLAN subinterface was configured statically
    - Dynamic—VLAN or S-VLAN subinterface was configured dynamically
  - Auto configure interface(s)—Types of dynamic upper interfaces configured with the **auto-configure** command: IP or PPPoE
  - Detected dynamic interface—Type of dynamic upper interface detected during autoconfiguration: IP, PPPoE, or (if no packet has been received) none
  - Interface types in lockout—Encapsulation types currently experiencing lockout: IP, PPPoE, or none
  - Lockout state (seconds)—Settings of encapsulation type lockout for the upper-layer encapsulation type indicated
    - Min—Minimum lockout time, in seconds
    - Max—Maximum lockout time, in seconds
    - Current—Current lockout time, in seconds; displays 0 (zero) if lockout is not occurring
    - Elapsed—Time elapsed into the lockout time, in seconds; displays 0 (zero) if lockout is not occurring
    - Next—Lockout time for the router to use for the next lockout event, in seconds

- In—Analysis of inbound traffic on this interface
  - Bytes—Number of bytes received on the VLAN or S-VLAN subinterface
  - Packets—Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all received packets; some packets might contain more than one error
  - Discards—Total number of discarded incoming packets
- Out—Analysis of outbound traffic on this interface
  - Bytes—Number of bytes sent on the VLAN or S-VLAN subinterface
  - Packets—Number of packets sent on the VLAN or S-VLAN subinterface
  - Multicast—Number of multicast packets received on the VLAN or S-VLAN subinterface
  - Broadcast—Number of broadcast packets received on the VLAN or S-VLAN subinterface
  - Errors—Total number of errors in all transmitted packets; some packets might contain more than one error
  - Discards—Total number of discarded outgoing packets
- ARP Statistics—Analysis of ARP traffic on this interface; In fields are for traffic received on the interface and Out fields are for traffic sent on the interface
  - ARP requests—Number of ARP requests
  - ARP responses—Number of ARP responses
  - Errors—Total number of errors in all ARP packets
  - Discards—Total number of discarded ARP packets
- Total VLAN interfaces—Total numbers of VLAN subinterfaces and VLAN major interfaces configured on the router; only this field appears when you specify the **summary** keyword
- Agent-Circuit-Identifier— Agent-circuit-identifier string

- Example 1—Displays full status and configuration information for all VLAN subinterfaces configured on the router

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype   Type
------------------------ ---------- ---- -------- -------- --------- ---------
ATM 3/0.1.2              Up         1522  ----    11       ----      Static
ATM 3/0.1.3              Up         1522  ----    12       ----      Static
ATM 3/1.1.1              Up         1522  ----    13       ----      Static
ATM 3/1.1.2              Up         1522  ----    14       ----      Static
ATM 3/2.1.1              Down       1526  4       255      0x9100    Static
FastEthernet 4/5.1       Up         1522  ----    1        ----      Dynamic
6 vlan subinterfaces found
```

- Example 2—Displays full status and configuration information for the specified VLAN subinterface

```
host1#show vlan subinterface fastEthernet 4/5.1
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 4/5.1       Up         1522 ----     1        ----      Dynamic
1 vlan subinterface found
```

- Example 3—Displays full status and configuration information for the specified S-VLAN ID

```
host1#show vlan subinterface svlan id 100 53
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 0/0.1       Up         1526 100      53       0x9100    Static
FastEthernet 4/6.1       Up         1526 100      53       0x9100    Dynamic

2 vlan subinterfaces found
```

- Example 4—Displays full status and configuration information for the specified dynamic VLAN subinterface

```
host1#show vlan subinterface fastEthernet 4/6.1000053
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype  Type
------------------------ ---------- ---- -------- -------- --------- -------
FastEthernet 4/6.1000053  Up         1526 100      53       0x9100    Dynamic

Auto configure interface(s)   : IP  PPPoE
Detected dynamic interface    : PPPoE
Interface types in lockout    : none

Lockout state (seconds)       : Min Max Current Elapsed Next
------------------------------ --- --- ------- ------- ----
IP                               1 300       0       0    1
PPPoE                            1 300       0       0    1

  In: Bytes 1040, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
  Out: Bytes 984, Packets 15
   Multicast 0, Broadcast 1
   Errors 0, Discards 0
ARP Statistics:
  In: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
  Out: ARP requests 0, ARP responses 0
   Errors 0, Discards 0
```

- Example 5—Displays status information for dynamic VLAN subinterfaces that are created based on agent-circuit-identifier information

```
host1#show vlan subinterface
        Interface           Status   MTU  Svlan Id Vlan Id  Ethertype   Type
------------------------ ---------- ---- -------- -------- --------- ---------
FastEthernet 4/0.1        Up         1522  2       ----     ----      Dynamic *
FastEthernet 4/0.2        Up         1522  2       ----     ----      Dynamic *
2 vlan subinterfaces found
* Created via agent circuit identifier

host1#show vlan subinterface agent-circuit-identifier
        Interface         Svlan Id Agent-Circuit-Identifier
------------------------ -------- --------------------------
FastEthernet 4/0.1        2        ----
FastEthernet 4/0.2        2        0200D0CB2729E5
```

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

X