Cisco 10000 Series Router Software Configuration Guide

August 2008

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-2226-19

CONTENTS

About This Guide

This guide provides configuration information for features that are platform-specific to the Cisco 10000 series router. Documentation is also provided for cross-platform features that function differently on the Cisco 10000 series router than on other supported platforms.

Cross-platform features that function on the Cisco 10000 series router as they do on other supported platforms, and platform-independent features that are supported on the Cisco 10000 series router are described in the general Cisco IOS documentation.

This introduction provides information about the following topics:

Guide Revision History

Cisco IOS Release	Part Number	Publication Date
Release 12.2(34)SB	OL-2226-19	August, 2008

Added the scaling limit of L4R sessions for PRE2, PRE3, and PRE4 in the Chapter 2, "Scalability and Performance"

Cisco IOS Release	Part Number	Publication Date
Release 12.2(33)SB	OL-2226-18	March, 2008

Removed Using PXF Commands and Configuring Layer 2 Local Switching chapters.

Added the features listed in the "New Features in Cisco 10000 Series Router Software Configuration Guide - IOS Release 12.2(33)SB" section on page 1-16.

Removed Using PXF Commands and Configuring Layer 2 Local Switching chapters.

Added the features listed in the "New Features in Cisco 10000 Series Router Software Configuration Guide - IOS Release 12.2(33)SB" section on page 1-16.

Cisco IOS Release	Part Number	Publication Date
Release 12.2(31)SB5	OL-2226-17	April, 2007

Added the GRE Tunnel IP Source and Destination VRF Membership feature in Chapter 24, "IP Tunneling."

Added the "New Features in Cisco IOS Release 12.2(31)SB5" section on page 1-17.

Cisco IOS Release	Part Number	Publication Date
Release 12.2(31)SB3	OL-2226-16	February, 2007

Description
Added the features listed in the "New Features in Cisco IOS Release 12.2(31)SB3" section on page 1-17.

Cisco IOS Release	Part Number	Publication Date
Release 12.2(31)SB2	OL-2226-15	November, 2006

Description
Added the features listed in the "New Features in Cisco IOS Release 12.2(31)SB2" section on page 1-18.

Cisco IOS Release	Part Number	Publication Date
Release 12.2(28)SB	OL-2226-14	July, 2006

Description
Added the features listed in the New Features in Cisco IOS Release 12.2(28)SB, page 1-19.

Cisco IOS Release	Part Number	Publication Date
Release 12.3(7)XI7	OL-2226-13	September, 2005

Description
Changed the Related Documentation link to the new *Cisco 10000 Series Router Documentation Roadmap*

Added the features listed in the "New Features in Cisco IOS Release 12.3(7)XI7" section on page 1-23.

Removed the "pointer to a pointer" for the PPPoE Circuit-Tag Processing feature by removing a summary and a pointer from Chapter 16, Configuring RADIUS Features, and retaining only the pointer to the feature module in the New Features in Cisco IOS Release 12.3(7)XI3, page 1-23.

Removed the restriction for non-support of SSG in Restrictions for IP Unnumbered on VLANs, page 7-3.

Added support for the 1-Port Channelized OC-12/STM-4 line card in Restrictions for MR-APS, page 14-3.

Removed Chapter 16, "IEEE 802.1Q-in-Q VLAN Tag Termination," and added a pointer to the *PPPoE—QinQ Support* feature guide, located at the following URL. This document includes support for IPoQ-in-Q.
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html

Relocated the remaining QoS features to the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

The chapter references for the following relocated features see the *Cisco 10000 Series Router Quality of Service Configuration Guide*:

- Class-based Weighted Fair Queuing—See "Sharing Bandwidth Fairly During Congestion"
- Define Interface Policy-Map AV Pairs AAA—See "Configuring Dynamic Subscriber Services"
- Dynamic Bandwidth Selection—See "Configuring Dynamic Subscriber Services"
- Hierarchical Shaping—See "Shaping Traffic"
- IP Quality of Service for Subscribers—See "Regulating Subscriber Traffic"
- MPLS QoS—See "Configuring Quality of Service for MPLS Traffic"
- MPLS Traffic Engineering—Diffserv Aware—See "Configuring Quality of Service for MPLS Traffic"
- Per VRF AAA (see Chapter 18, "Configuring Quality of Service for MPLS Traffic")

Added feature histories and mini tables of contents for each feature in this guide.

Added the Static MAC Address for PPPoE feature in Chapter 6, "Configuring PPPoE over Ethernet and IEEE 802.1Q VLAN"

Cisco IOS Release	Part Number	Publication Date
Release 12.3(7)XI6	OL-2226-10	June, 2005

Description

Corrected MR-APS configuration in Example 14-1.

Added output policing behavior on an LNS VAI (CSCee07016) in Restrictions for the LNS, page 5-28.

Corrected examples to show VLANs instead of ATM PVCs in Chapter 6, "Configuring PPPoE over Ethernet and IEEE 802.1Q VLAN".

Added a chapter to describe frequently-used show PXF commands in Chapter 23, "Using PXF Commands".

Revised a note about mapping sessions to VRFs by using the RADIUS server in PPP over Ethernet to MPLS VPN, page 3-5.

Added a description of PRE support on Cisco 10000 series routers in Hardware Requirements, page 1-1.

Cisco IOS Release	Part Number	Publication Date
Release 12.3(7)XI3	OL-2226-09	March, 2005

Description

Added the features listed in the "New Features in Cisco IOS Release 12.3(7)XI3" section on page 1-23.

Corrected scaling limits for active VCs on ATM line cards (CSCeg37235) in:

- VC Scaling, page 2-7
- Configuring atm pxf queuing, page 2-16
- Restrictions for Hierarchical Shaping (moved to the *Cisco 10000 Series Router Quality of Service Configuration Guide*)
- ATM VC Scaling and VC Assignment, page 8-4
- Restrictions for VBR-nrt Oversubscription, page 8-15

Changed the configurable ATM oversubscription factor range from 1-50 to 1-500 in Configuring VBR-nrt Oversubscription, page 8-17

Corrected the restrictions for MPLS QoS to indicate that the **set mpls experimental imposition topmost** command is not supported.

Added a restriction for enabling IP multicast fast switching in Restrictions for IP Multicast, page 15-2

Changed the title of this guide to include MPLS configuration

Relocated QoS features to the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

The chapter references in the following relocated features see the *Cisco 10000 Series Router Quality of Service Configuration*:

- Modular QoS CLI Overview—See "Quality of Service Overview."
- MQC Policy Map Support on Configured VC Range ATM—See "Attaching Service Policies."
- Strict Priority Queuing—See "Prioritizing Services."
- 3-Color Policer—See "Policing Traffic."
- Percent-Based Policing—See "Policing Traffic."
- Queue Scaling—See "Managing Packet Queue Congestion."
- IEEE 802.1p Class of Service—See "Marking Traffic."
- Per DSCP Weighted Random Early Detection—See "Managing Packet Queue Congestion."
- Per Precedence Weighted Random Early Detection Statistics—See "Managing Packet Queue Congestion."
- Weighted Random Early Detection with Queue Limit—See "Managing Packet Queue Congestion."
- VC Weighting—See "Oversubscribing Physical and Virtual Links."
- Dynamic ATM VP and VC Configuration Modification—See "Oversubscribing Physical and Virtual Links."
- Interface Oversubscription—See "Oversubscribing Physical and Virtual Links."

- 3-Level Hierarchical QoS Policies—See “Defining QoS for Multiple Policy Levels.”

Cisco IOS Release	Part Number	Publication Date
Release 12.3(7)XI2	OL-2226-08	November, 2004

Description
Added the features listed in the “New Features in Cisco IOS Release 12.3(7)XI2” section on page 1-24.

Added a scaling limitation for create on demand PVCs and PPP sessions in Limitations and Restrictions, page 2-3

Changed the SAR page limit (CSCee59870) in ATM VC Scaling and VC Assignment, page 8-4

Added information about the behavior of high water mark and low water mark values used with VC weighting in High Water Mark and Low Water Mark Values (moved to the *Cisco 10000 Series Router Quality of Service Configuration Guide*)

Added a table indicating scaling limits for active VCs on ATM line cards in:

- Configuring atm pxf queuing, page 2-16
- Restrictions for Hierarchical Shaping (moved to the *Cisco 10000 Series Router Quality of Service Configuration Guide*)
- ATM VC Scaling and VC Assignment, page 8-4
- Restrictions for VBR-nrt Oversubscription, page 8-15

Cisco IOS Release	Part Number	Publication Date
Release 12.3(7)XI1	OL-2226-07	August, 2004

Description
Added the new features listed in the “New Features in Cisco IOS Release 12.3(7)XI1” section on page 1-24.

Audience

This guide is designed for system and network managers responsible for configuring broadband aggregation, leased-line, and MPLS services and on the Cisco 10000 series router. The manager should be experienced using Cisco IOS software and be familiar with the operation of the Cisco 10000 series router.

Document Organization

This guide contains the following chapters:

Chapter	Title	Description
Chapter 1	Broadband Aggregation Overview	Lists new features and enhancements in each release; describes hardware requirements. Provides examples of broadband and leased-line architecture models.
Chapter 2	Scalability and Performance	Describes limitations and restrictions, and how to configure the Cisco 10000 series router for high scalability.
Chapter 3	Configuring Remote Access to MPLS VPN	Describes the Remote Access (RA) to MPLS VPN feature that allows the service provider to offer a scalable end-to-end VPN service to remote users.
Chapter 4	Configuring Multiprotocol Label Switching	Describes MPLS-related features, such as BGP Multipath load sharing, Session Limit per VRF, and Half-duplex VRF.
Chapter 5	Configuring Layer 2 Tunnel Protocol Access Concentrator and Network Server	Describes how to configure the Cisco 10000 series router as a Layer 2 Tunnel Protocol Access Concentrator (LAC) or as an L2TP Network Server (LNS). The managed LNS feature of the Cisco 10000 series router enables the router to assign a subscriber session to a VRF instance and route the session within the VRF to the destination network.
Chapter 6	Configuring PPPoE over Ethernet and IEEE 802.1Q VLANs	Describes the PPPoE over Ethernet feature that enables direct connection to an Ethernet interface. Also describes the IEEE 802.1Q VLANs feature that enables the Cisco 10000 series router to support PPPoE over IEEE 802.1Q encapsulated VLANs using Gigabit Ethernet.
Chapter 7	Configuring IP Unnumbered over VLAN	Describes the IP Unnumbered over VLAN feature that helps service providers to conserve IP address space for service provider configurations that include Ethernet VLAN subinterfaces.
Chapter 8	Configuring ATM Permanent Virtual Circuit Autoprovisioning	Describes how to configure the ATM PVC autoprovisioning feature that enables DSL wholesale service providers to dynamically provision ATM service for subscribers using a local configuration. Also describes the VBR-nrt Oversubscription feature.
Chapter 9	Configuring the Multihop Feature	Describes how to configure the multihop feature that enables the Cisco 10000 series router to terminate sessions arriving in L2TP tunnels from LACs and to forward the sessions through new L2TP tunnels to the router's peer L2TP Network Server (LNS). Also describes how to configure the preservation of the IP type of service (ToS) field for tunneled IP packets.
Chapter 10	Configuring Address Pools	Describes address assignment mechanisms, including the on-demand address pool manager feature and the overlapping addresses feature. Describes how to configure each of these features.
Chapter 11	Configuring Local AAA Server, User Database—Domain to VRF	Describes the Local AAA Server, User Database—Domain to VRF feature, which extends the Cisco IOS AAA Authorization to local AAA profiles on the router without using an AAA Server.

Chapter	Title	Description
Chapter 12	Configuring Traffic Filtering	Describes the IP Receive ACLs and Time-Based ACLs features that provide filtering capability for traffic that is destined for the router and protects the router from remote intrusions.
Chapter 13	Unicast Reverse Path Forwarding	Describes the Unicast Reverse Path Forwarding feature that verifies if the path of an incoming packet is consistent with the local packet forwarding information. The validity of this path determines whether uRPF passes or drops the packet.
Chapter 14	Configuring Automatic Protection Switching	Describes the Multirouter Automatic Protection Switching (MR-APS) feature that enables SONET connections to switch from one SONET circuit to another SONET circuit if a circuit failure occurs.
Chapter 15	Configuring IP Multicast	Describes the IP Multicast feature.
Chapter 16	Configuring RADIUS Features	Describes the RADIUS attribute screening, RADIUS transmit retries, RADIUS Attribute 31: PPPoX Calling-Station-ID, and RADIUS packet of disconnect features.
Chapter 17	Configuring L2 Virtual Private Networks	Describes L2VPN features of both LS and AToM types available on Cisco 10000 series router.
Chapter 18	Configuring L2VPN Interworking	Describes L2 interworking features available on Cisco 10000 series router.
Chapter 19	Configuring Multilink Point-to-Point Connections	Describes MLP and how to configure it on serial and ATM connections on the Cisco 10000 series router.
Chapter 20	Configuring Gigabit EtherChannel Features	Describes Gigabit EtherChannel features available on Cisco 10000 series router.
Chapter 21	Configuring IP Version 6	Lists the IPv6 features that are supported on the Cisco 10000 series router and notes limitations of that support.
Chapter 22	Configuring Template ACLs	Describes Template ACLs, in which one ACL represents many similar ACLs.
Chapter 23	Protecting the Router from DoS Attacks	Describes how to protect against denial of service (DoS) attacks.
Chapter 24	IP Tunneling	Describes the Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership feature.
Appendix A	RADIUS Attributes	Lists RADIUS attributes that the Cisco 10000 series router supports.

This guide also includes a Glossary and an Index.

Document Conventions

This guide uses the following conventions:

- **Bold** is used for commands, keywords, and buttons.
- *Italics* are used for command input for which you supply values.
- `Screen` font is used for examples of information that are displayed on the screen.

- **`Bold screen`** font is used for examples of information that you enter.
- Vertical bars (|) indicate separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces ({}) indicate a required choice.
- Braces within square brackets ([{}]) indicate a required choice within an optional element.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the guide.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning **Means *danger*. You are in a situation that could cause bodily injury. Before you work on any equipment, you must be aware of the hazards involved with electrical circuitry and familiar with standard practices for preventing accidents. To see translated versions of warnings, refer to the *Regulatory Compliance and Safety Information* document that accompanied the device.**

Related Documentation

For more information about the Cisco 10000 series router, its features, and hardware, go to the Cisco 10000 series router documentation roadmap, located at the following URL:

http://www.cisco.com/en/US/products/hw/routers/ps133/products_documentation_roadmap09186a00804ba4f3.html

For information about Cisco IOS Release 12.2, including command reference and system error messages, go to the Cisco IOS Release 12.2 documentation web page, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/tsd_products_support_series_home.html

RFCs

RFC	Title
RFC 791	Internet Protocol
RFC 1163	A Border Gateway Protocol (BGP)
RFC 1483	Multiprotocol Encapsulation over ATM
RFC 1490	Multiprotocol Interconnect over Frame Relay
RFC 1661	The Point-to-Point Protocol (PPP)
RFC 1990	The PPP Multilink Protocol (MP)
RFC 2373	IP Version 6 Addressing Architecture
RFC 2516	A Method for Transmitting PPP Over Ethernet (PPPoE)
RFC 2529	Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
RFC 2661	Layer Two Tunneling Protocol "L2TP"
RFC 2685	Virtual Private Networks Identifier
RFC 2867	RADIUS Accounting Modifications for Tunnel Protocol Support
RFC 3036	LDP Specification
RFC 3107	Carrying Label Information in BGP-4
RFC 3587	IPv6 Global Unicast Address Format
RFC 4193	Unique Local IPv6 Unicast Addresses
RFC 4659	BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

CHAPTER 1

Broadband Aggregation and Leased-Line Overview

The Cisco 10000 series router is a highly scalable and reliable IP edge platform, providing nonstop performance for service providers deploying IP services. With the rapid growth in broadband customers, the Cisco 10000 series router accommodates the service provider's need for an expanding set of broadband aggregation features.

This chapter provides an overview of the broadband aggregation features available on the Cisco 10000 series router and includes the following topics:

Hardware Requirements

The performance routing engine (PRE) performs all Layer 2 and Layer 3 packet manipulation related to routing and forwarding operations. Table 1-1 shows PRE support on Cisco 10000 series routers.

Table 1-1 PRE Support on Cisco 10000 Series Routers

Performance Routing Engine Support				
Chassis	**ESR-PRE**	**PRE1**	**PRE2**	**PRE3**
Cisco 10005	Yes	Yes	No	No
Cisco 10008	Yes	Yes	Yes	Yes

Checking Hardware and Software Compatibility

The PRE installed in the Cisco 10000 series router chassis must support the Cisco IOS software running on the router. Use the **show version** command to check the PRE version installed.

To see if a feature is supported by a Cisco IOS release, to locate the software document for that feature, or to check the minimum software requirements of Cisco IOS software with the hardware installed on your router, Cisco maintains the Software Advisor tool on Cisco.com at http://www.cisco.com/cgi-bin/Support/CompNav/Index.pl

You must be a registered user on Cisco.com to access this tool.

Broadband Architecture Models

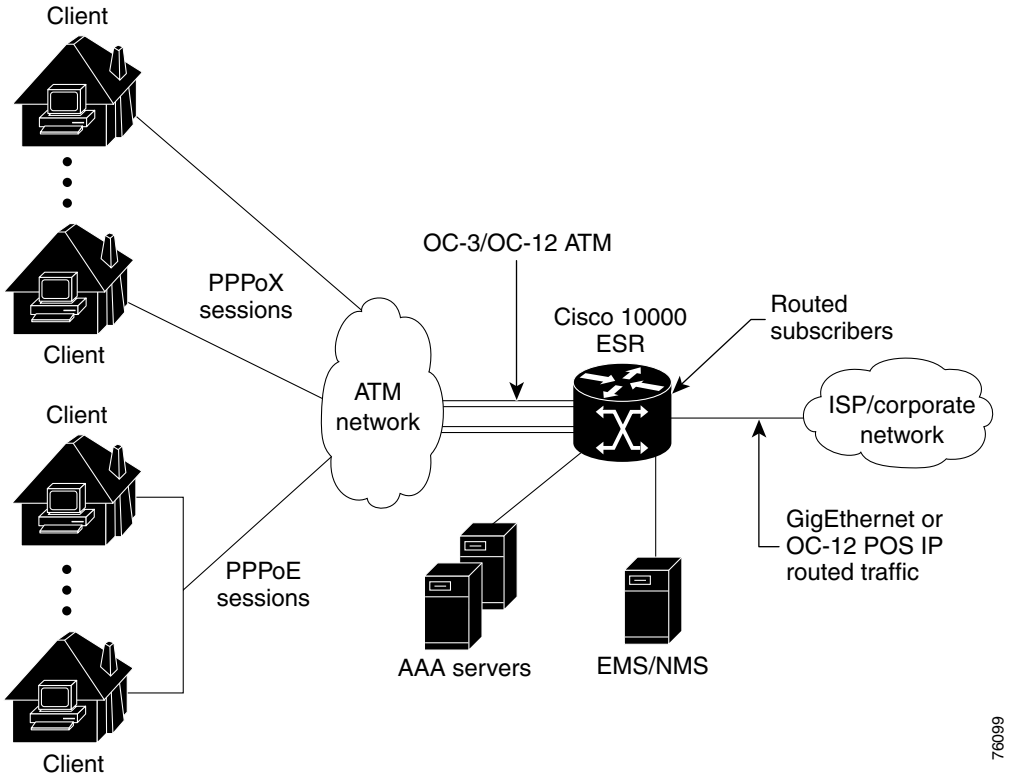
This section shows broadband models for the following architectures:

- PPP termination and aggregation (PTA) for PPPoA or PPPoE
- PTA to virtual routing and forwarding (VRF)
- PTA to Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN)
- L2TP network server (LNS)
- L2TP to VRF
- L2TP over MPLS to VRF
- L2TP access concentrator (LAC)
- Routed bridge encapsulation (RBE)
- RBE to VRF
- RBE to MPLS VPN

PPP Termination and Aggregation Architectures

Figure 1-1 shows a PPP termination and aggregation (PTA) model for PPP over ATM (PPPoA) or PPP over Ethernet (PPPoE) sessions.

Figure 1-1 PTA Architectural Model



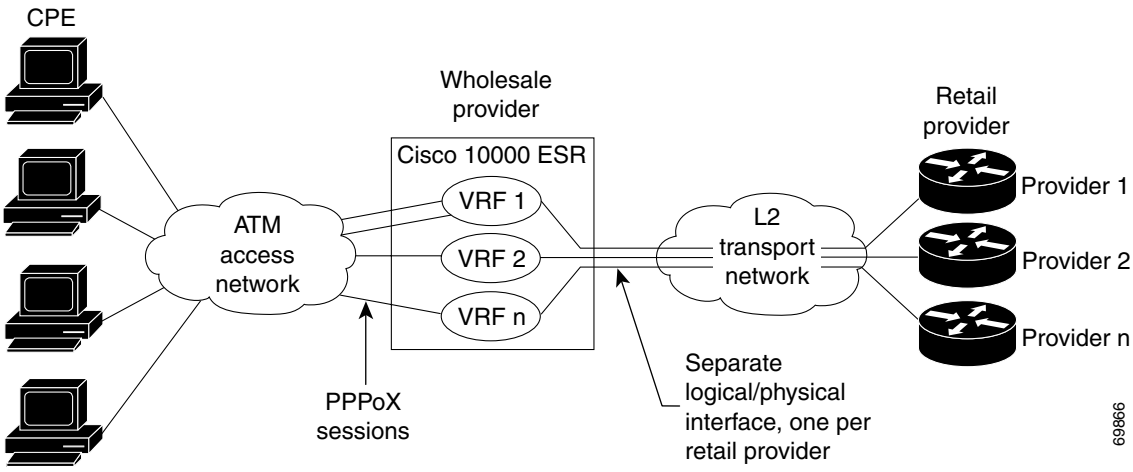
In the figure, an ATM network (with no routing capability) is between the clients and the Cisco 10000 series router. Each client session arrives on a VC (multiple sessions and PCs can use this single VC). The IP traffic of the client is encapsulated in PPPoX. The Cisco 10000 series router terminates the PPP sessions and routes the client data packets toward their final destination, typically onto the ISP or corporate network.

Note PPPoX refers to either PPPoA or PPPoE.

PTA to Virtual Routing and Forwarding Architecture

Figure 1-2 shows a PPP termination and aggregation (PTA) to virtual routing and forwarding (VRF) model for PPPoA or PPPoE sessions.

Figure 1-2 PTA to VRF Architectural Model

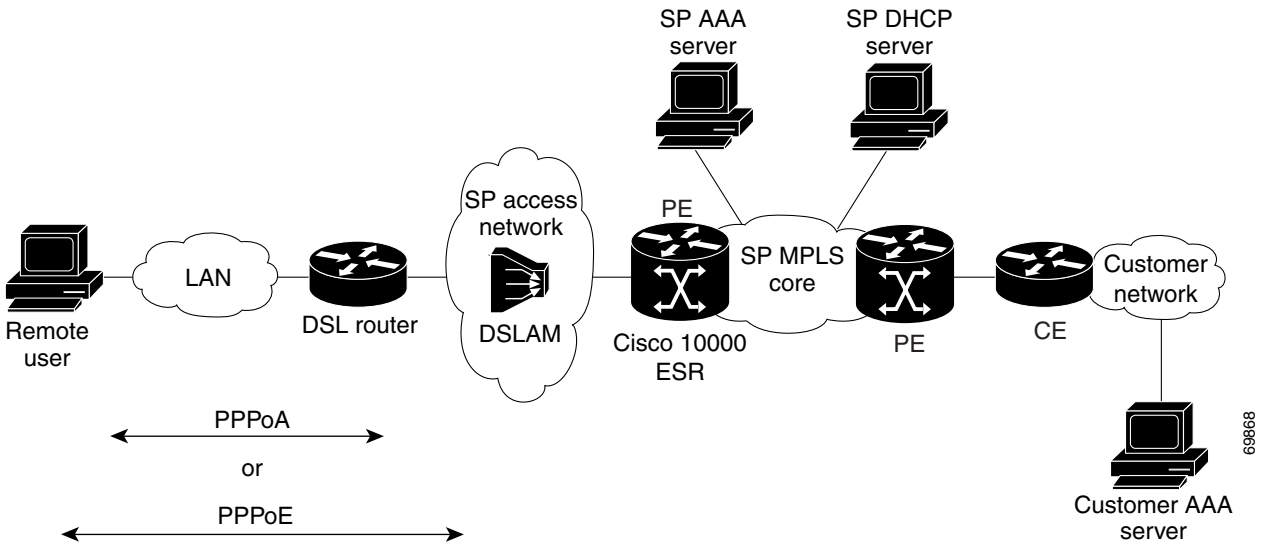


In this model, the Cisco 10000 series router terminates the sessions and places the sessions in the appropriate VRF. This model is identical to the one in Figure 1-3 on the access side. However, the two models differ on the network side. The model in Figure 1-2 uses VRFs, does not use a tag interface on the network side, and separates traffic at Layer 2. The "PTA to MPLS VPN Architectural Model" in Figure 1-3 uses MPLS and a tag interface, and separates traffic at Layer 3.

PTA to Multiprotocol Label Switching Virtual Private Network Architecture

Figure 1-3 shows a MPLS VPN model for PPPoA or PPPoE sessions.

Figure 1-3 PTA to MPLS VPN Architectural Model

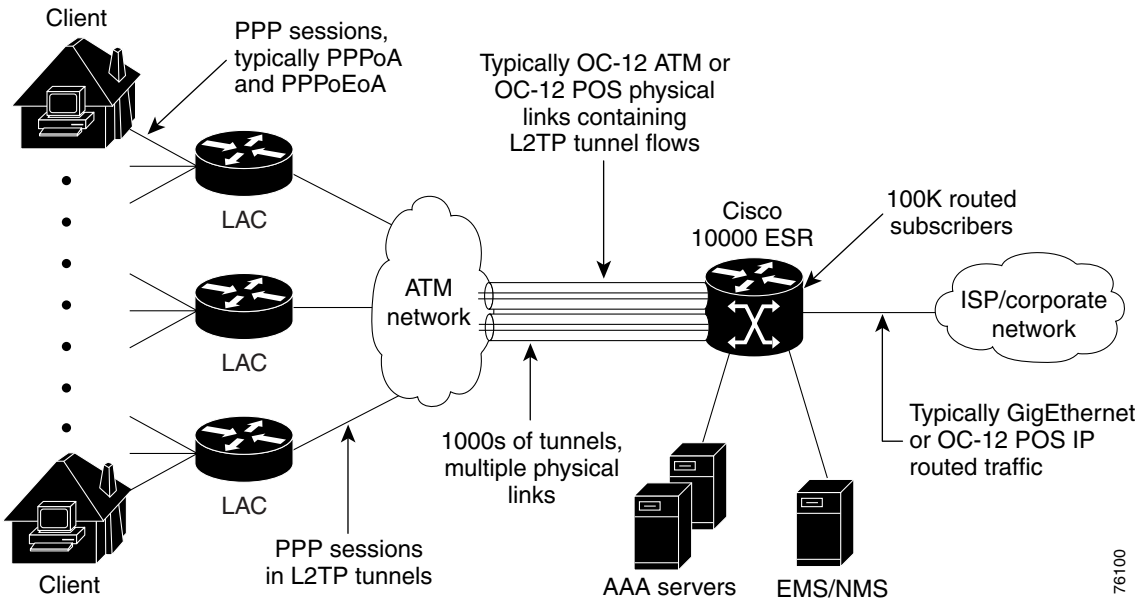


In the figure, PPPoX sessions are placed in the proper virtual routing and forwarding (VRF) instance based on the virtual template to which they map. This model is identical to the one in Figure 1-2 on the access side. However, the two models differ on the network side. The model in Figure 1-3 uses MPLS and a tag interface on the network side and separates traffic at Layer 3. The "PTA to VRF Architectural Model" in Figure 1-2 uses VRFs does not use a tag interface, and separates traffic at Layer 2.

L2TP Architectures

Figure 1-4 shows an L2TP network server (LNS) model.

Figure 1-4 LNS Architectural Model

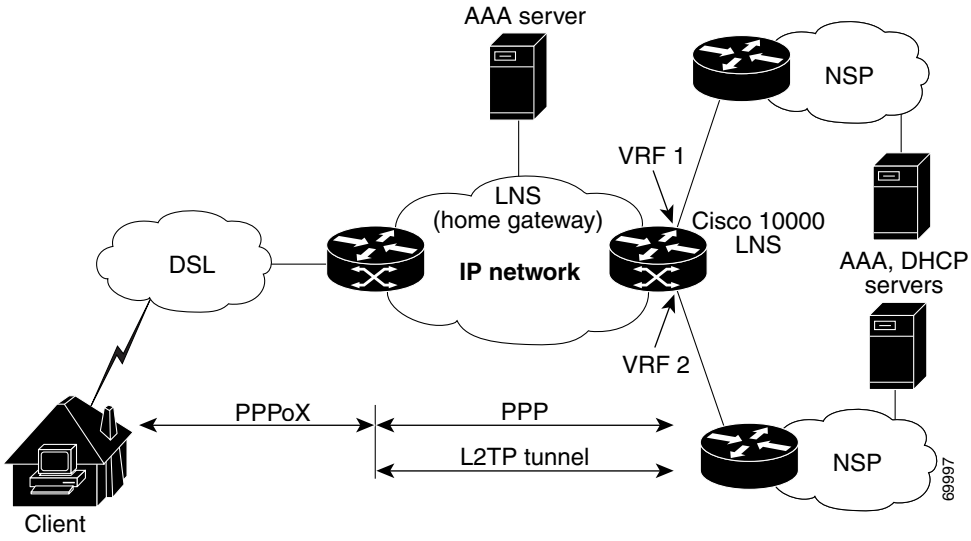


In the figure, the clients and the LACs exchange PPP packets that are typically encapsulated in PPPoA or PPPoE and typically carried on ATM circuits. However, the protocols used between the clients and the LAC do not affect LNS requirements. The LAC creates L2TP tunnels to all of the LNSs at which its clients want to terminate. Multiple tunnels might exist between each LAC and each LNS. For each client PPP session the LAC signals the LNS to add another session to a tunnel. The LAC forwards all traffic to the LNS, including the PPP control traffic. The LNS terminates the PPP sessions and routes any client IP packets on to the ISP or corporate network toward their final destination. The LNS performs authentication, authorization, and accounting (AAA) actions on the PPP sessions.

L2TP to Virtual Routing and Forwarding Architecture

Figure 1-5 shows an L2TP to VRF model.

Figure 1-5 L2TP to VRF Architectural Model

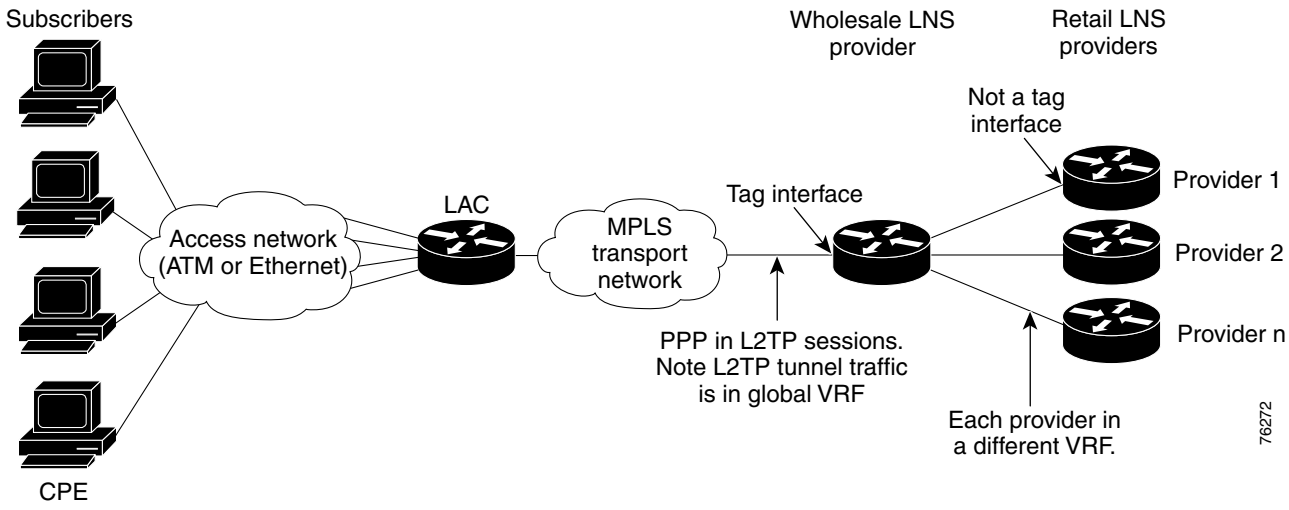


In this model, the Cisco 10000 series router acts as the LNS with VRF 1 and VRF 2 configured on the router. PPPoX sessions are placed in an L2TP tunnel and terminated at the LNS where they are placed in the appropriate VRF.

L2TP over MPLS to Virtual Routing and Forwarding Instance

Figure 1-6 shows PPP in L2TP tunneled traffic transported over an MPLS tag interface to the wholesale LNS provider.

Figure 1-6 L2TP over MPLS to VRF Architectural Model

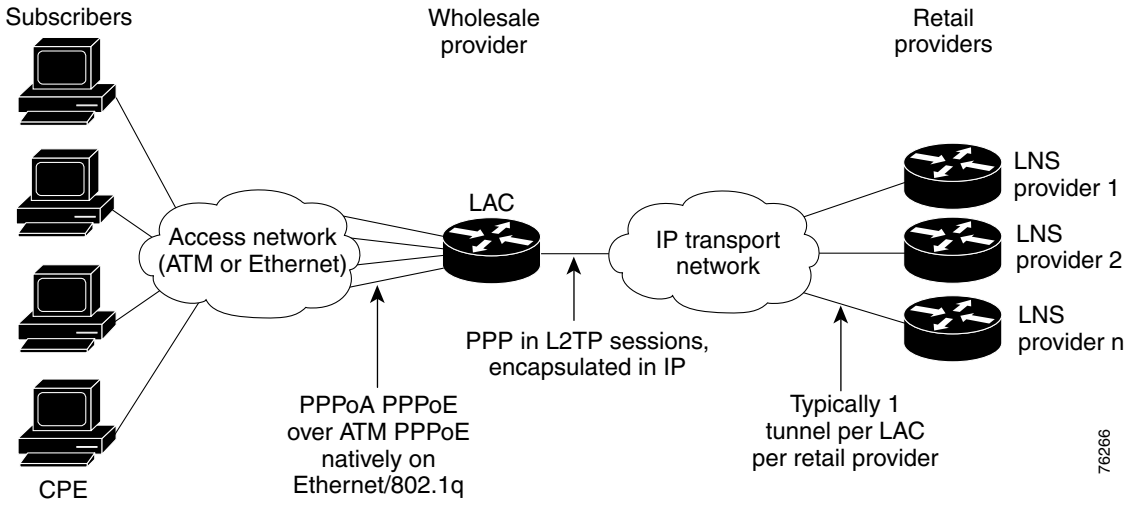


The LNS encapsulates the PPP in L2TP sessions in IP packets and forwards them to the retail LNS providers, placing the sessions for each provider in separate VRFs.

L2TP Access Concentrator Architecture

Figure 1-7 shows an L2TP access concentrator (LAC) model.

Figure 1-7 LAC Topology

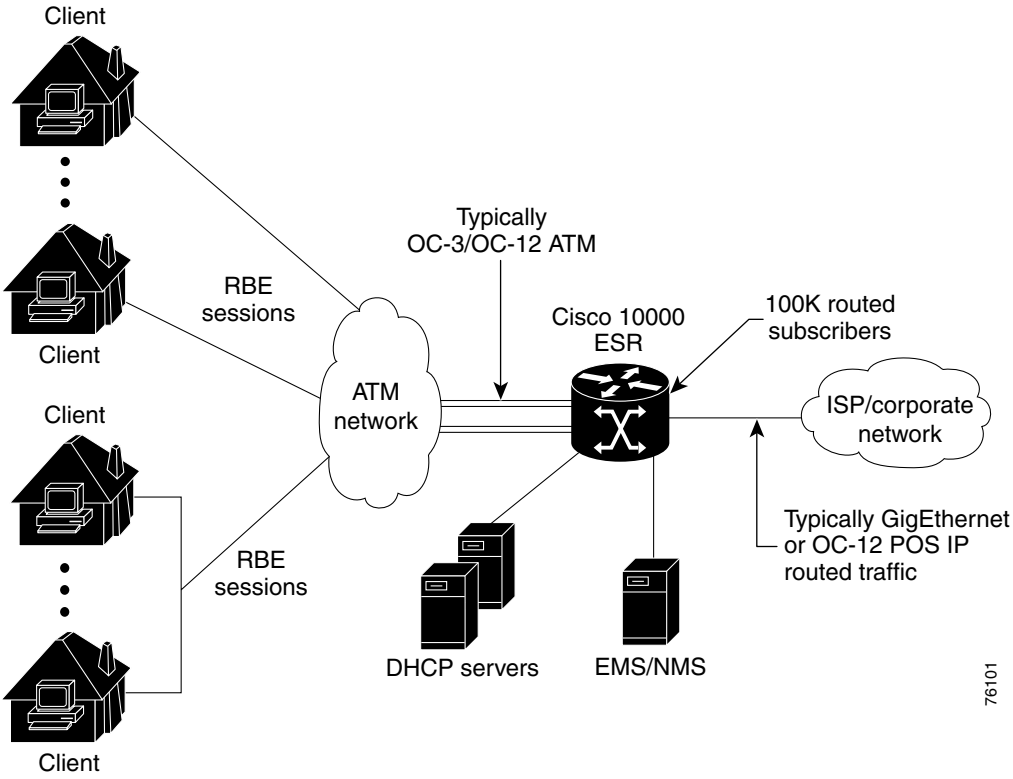


In the figure, wholesale providers tunnel subscriber PPP sessions to the retail provider. PPP in L2TP sessions are encapsulated in IP packets and forwarded over any IP transport network.

Routed Bridge Encapsulation Architectures

Figure 1-8 shows a routed bridge encapsulation (RBE) model.

Figure 1-8 RBE Architectural Model

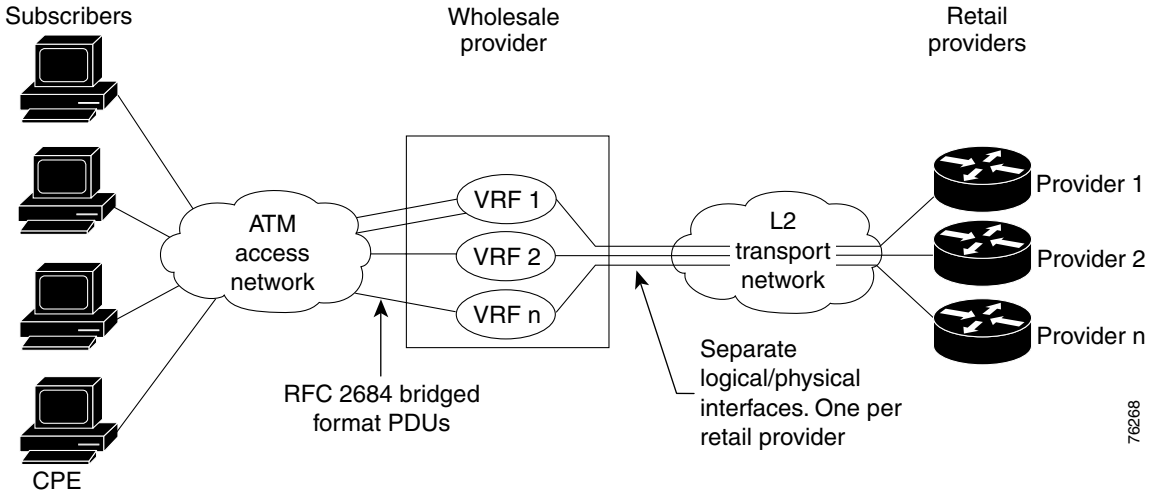


In the figure, an ATM network (with no routing capability) is between the clients and the Cisco 10000 series router. Each client session arrives on a VC (multiple sessions and PCs can use this single VC). IP traffic of the client is encapsulated in RBE. The Cisco 10000 series router processes ARP or DHCP requests and routes the client data packets toward their final destination, typically onto the ISP or corporate network.

RBE to Virtual Routing and Forwarding Architecture

Figure 1-9 shows an RBE to VRF model.

Figure 1-9 RBE to VRF Topology

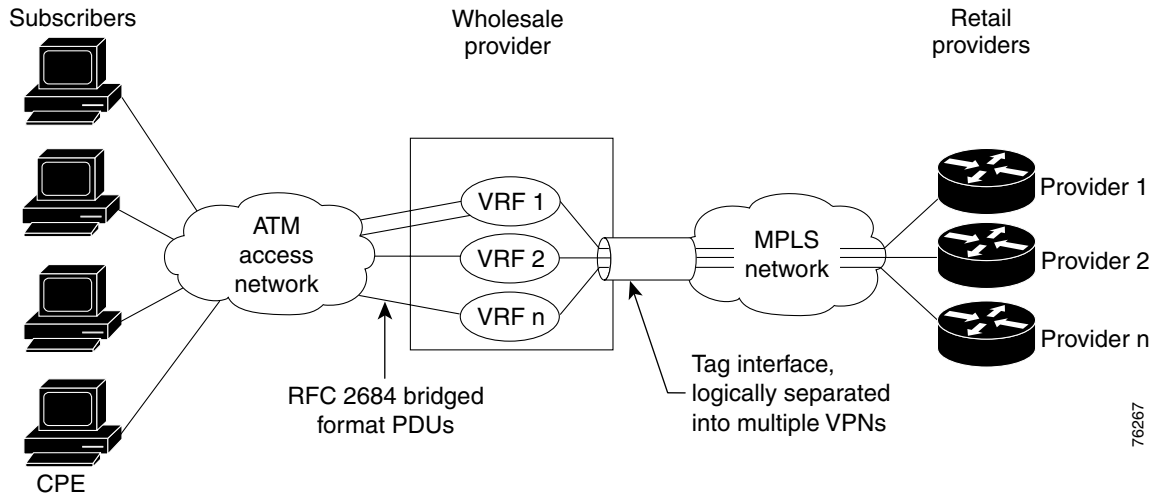


In the figure, the wholesale provider uses physical or logical interfaces to separate the subscribers of different retail providers. On the access side, the subscribers are uniquely placed in VRFs. A separate physical or logical interface to each retail provider separates traffic for the different retail providers on the network side.

RBE to Multiprotocol Label Switching Virtual Private Network Architecture

Figure 1-10 shows an RBE to MPLS VPN model.

Figure 1-10 RBE to MPLS VPN Topology



In the figure, the wholesale provider uses VPNs to separate the subscribers of different retail providers. On the access side, the subscribers are uniquely placed in VRFs. A tag interface separates traffic for the different retail providers on the network side. The MPLS VPN technology is used to assign tags in a VPN aware manner.

Leased-Line Architecture Models

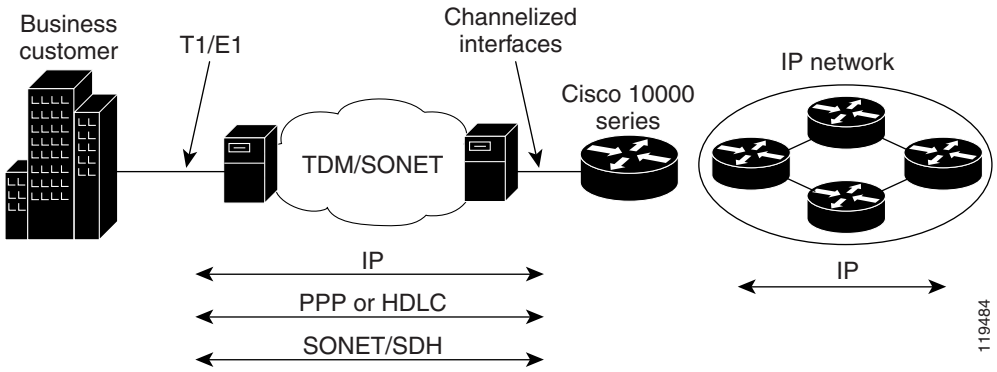
This section shows leased-line models for the following architectures and applications:

- Channelized aggregation
- Frame Relay aggregation
- ATM aggregation
- Ethernet aggregation
- MPLS provider edge application
- Combined Broadband and Leased-Line applications

Channelized Aggregation

The Cisco 10000 series router allows the aggregation of low-speed, very-high-density leased-line circuits by using channelized interfaces. Figure 1-11 shows an example of channelized architecture.

Figure 1-11 Channelized Architecture



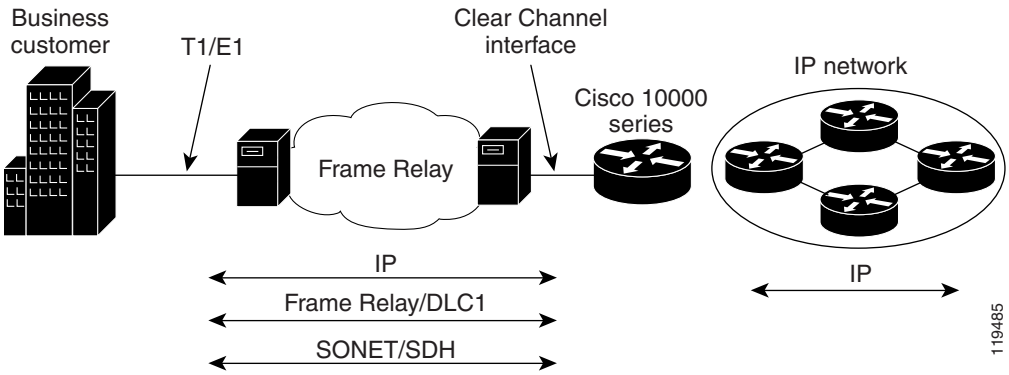
In a typical Cisco 10000 series router application, the provider usually situates the aggregator in a centrally located POP and backhauls individual customer connections from central offices across the SONET/SDH networks. Add-drop multiplexers at either end of the optical network that provide aggregation of low-speed customer connections (T1/E1) and aggregation into higher-order optical interfaces in the central POP. Numerous IP services are supported over channelized interfaces, including IP QoS, ACLs, IP multicast, and security services.

Frame Relay Aggregation

Many service providers offer IP Internet access and VPN products over existing Frame Relay access networks. Frame Relay packet-switched networks allow flexibility to allocate resources based on traffic profiles. When aggregating Frame Relay circuits, the Cisco 10000 series router is usually located in a central POP and connects to local switch nodes through copper or optical interfaces. Typically, these connections are implemented with nonchannelized interfaces. Frame Relay data-link connection identifiers (DLCIs) are terminated on the Cisco 10000 series router with customer IP traffic routed through the core network. Frame Relay encapsulation is supported on many interfaces, including channelized and nonchannelized modules. Numerous Frame Relay options and services are supported on the platform, including traffic shaping and QoS.

Figure 1-12 shows an example of Frame Relay architecture.

Figure 1-12 Frame Relay Architecture

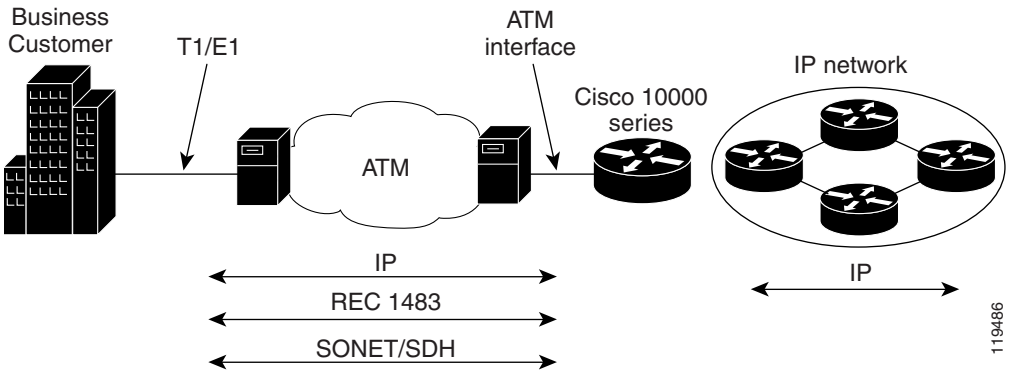


ATM Aggregation

ATM is used in many local exchange carrier (ILEC) and PTT access networks, and many providers use the technology as the foundation for multiservice platforms. ATM can be used to provide transport services for many applications, including backhaul for DSL services and leased-line emulation for Internet and VPN services.

Figure 1-13 shows an example of ATM architecture.

Figure 1-13 ATM Architecture



When used as an ATM aggregator, the Cisco 10000 series router is usually placed in a central POP and connected to a local ATM switching node through optical interfaces. ATM virtual circuits are terminated on the device, and customer IP traffic destined for the Internet or VPN is routed onto the core network.

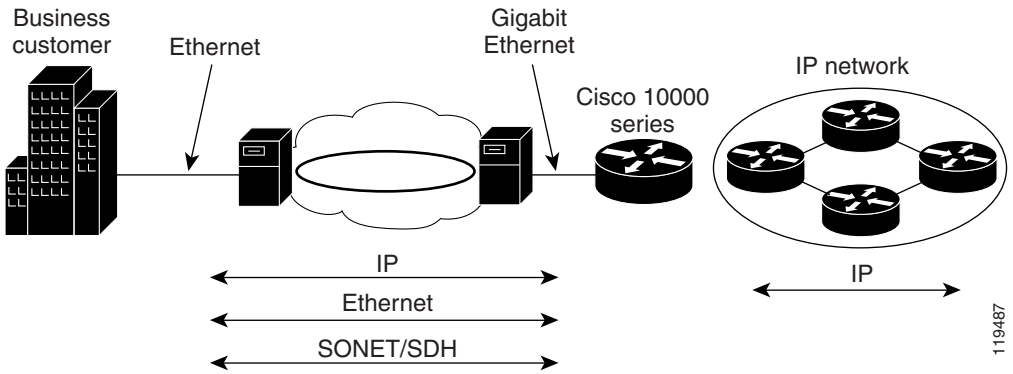
The Cisco 10000 series router supports ATM classes of service (CoSs), including UBR, UBR+, VBR-nrt, and CBR with extensive IP QoS to ATM CoS interworking. The ATM feature set includes accurate and scalable traffic shaping as well as operations, administration, and maintenance (OAM) facilities.

Ethernet Aggregation

Many enterprise customers use Ethernet technology for the "hub" site within a VPN network. "Spoke" sites are generally connected to the service provider infrastructure with lower speed fixed circuits. Customer connections are usually defined as 802.1Q virtual LAN (VLAN) logical interfaces under the main Ethernet interface. The Cisco 10000 series router supports both Gigabit and Fast Ethernet interfaces with many IP services, including QoS and ACLs.

Figure 1-14 shows an example of Ethernet architecture.

Figure 1-14 Ethernet Architecture

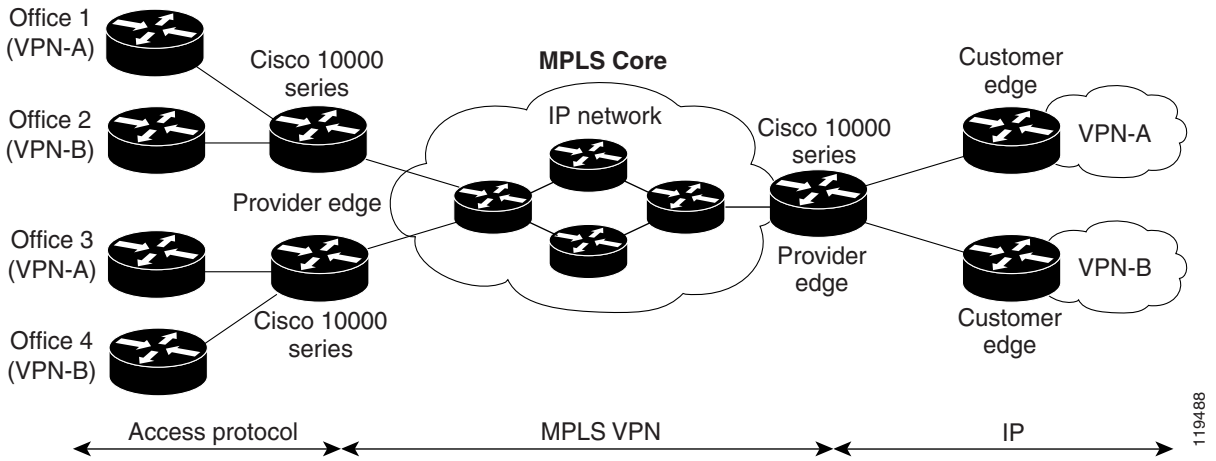


MPLS Provider Edge Applications

MPLS technology has allowed providers to target small to medium-sized businesses for outsourced VPN services. The "build once, sell many" approach of the network design provides scalability and flexibility with respect to VPN products and services. MPLS provider edge functions and associated features and services are offered on the Cisco 10000 series router, spanning all interfaces and encapsulations from low-speed broadband to traditional leased-line applications to high-speed Ethernet.

Figure 1-15 shows an example of MPLS architecture.

Figure 1-15 MPLS Architecture

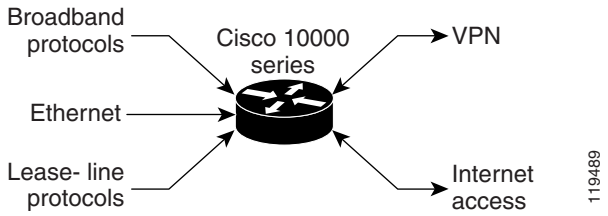


Combined Broadband and Leased-Line Applications

The demarcation between leased-line and broadband applications has become less clear over the past few years. DSL circuits are competing in the traditional leased-line space, with many service providers offering Internet and VPN services over these lower-cost alternatives to dedicated TDM. The role of the leased-line aggregator has expanded to include the termination of many traditional broadband interfaces and encapsulations. Combining leased-line and business-class DSL access is one option that many providers are introducing to reduce costs and consolidate the number of edge products.

Figure 1-16 shows an example of combined broadband and leased-line architecture.

Figure 1-16 Combined Broadband and Leased-Line Architecture



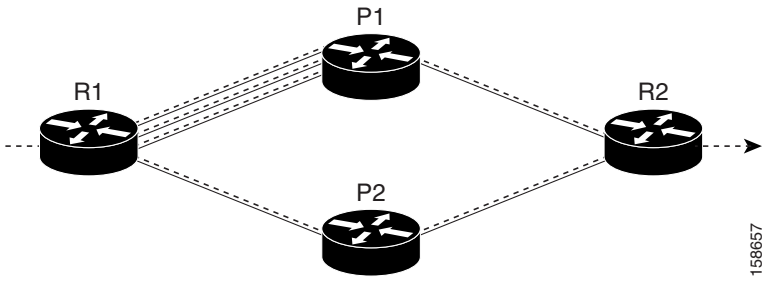
Load Balancing Architecture Models

This section describes how the Cisco 10000 series router load balances traffic in various network topologies. The scenarios apply to a Cisco 10000 series router with a PRE2.

IP and MPLS Applications

Figure 1-17 shows a simple network topology that uses IP or basic MPLS forwarding. It does not include MPLS VPN routes. There are multiple outgoing paths from the R1 router to the R2 router. Load balancing is achieved by populating multiple paths in the PXF. On a Cisco 10000 series router, load balancing is supported on a maximum of eight unique paths.

Figure 1-17 IP and MPLS Load Balancing



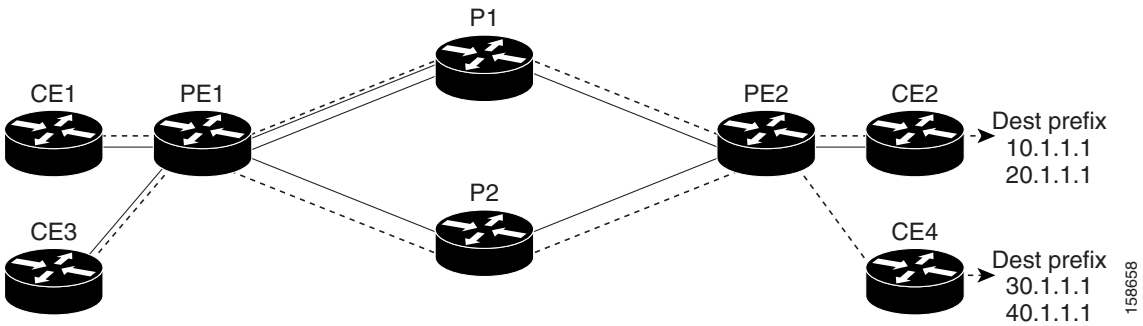
You can set load balancing to work per-destination or per-packet. For per-destination load balancing, the packet arrives at R1 and the hash value is computed based on the source IP address, destination IP address, and router ID. The PXF has a proprietary algorithm to select a path based on the number of total paths available.

Per-packet load balancing allows data traffic to be evenly distributed in an IP network over multiple equal-cost connections. Per-packet load balancing uses round-robin techniques to select the output path without basing the choice on the packet content.

Single Ingress and Single Egress Provider Edge Applications

Figure 1-18 shows the provider edge 1 (PE1) router with three Interior Gateway Protocol (IGP) routes into the core. Load balancing from customer edge 1 (CE1) to CE2 occurs on the PE1 router into different paths. There is a single path for all destination prefixes on CE2 and a separate path for all destination prefixes on CE4.

Figure 1-18 Single Ingress and Single Egress PE Load Balancing



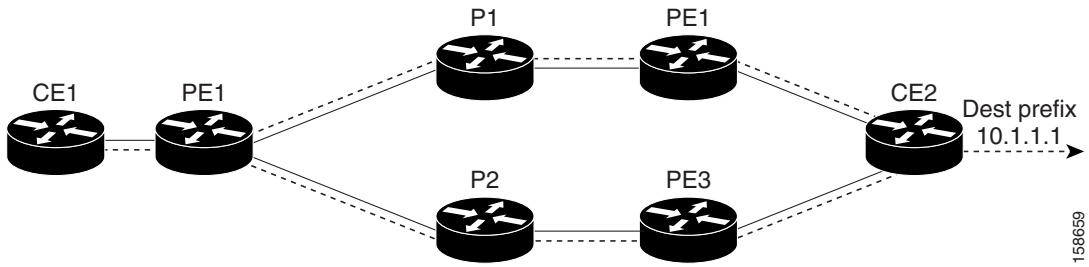
For each destination prefix on a destination CE that requires unique Label Switched Path (LSP), selection of the outgoing IGP path is in round-robin fashion. When there are multiple IGP paths from the ingress PE to egress PE, the outgoing IGP path is chosen statically upon processing by the PXF. For different destination prefixes, path selection is round-robin and each destination prefix has only one path. All destination IP addresses mapping to the same destination prefix take the same path.

When there are multiple destination prefixes, load balancing occurs on traffic across the IGP paths. In the case of only one or a few destination prefixes, load balancing does not occur on traffic across the IGP paths and this behavior is the same whether load balancing is configured per-destination or per-packet.

Single Ingress and Two Egress Provider Edge Applications

Figure 1-19 shows the routing of packets from CE1 to CE2 using the PE1 router. There are multiple paths for the destination prefixes on CE2. Load balancing occurs in the PXF of PE1.

Figure 1-19 Single Ingress and Two Egress PE Load Balancing



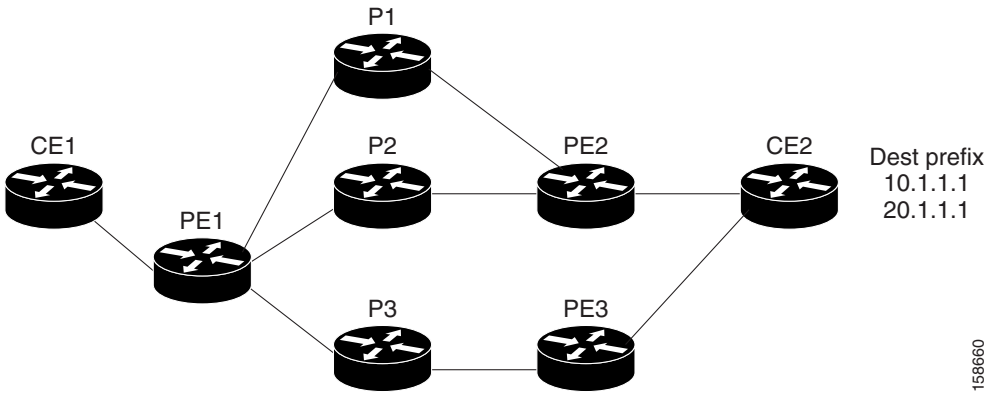
You can set load balancing to work per-destination or per-packet. For per-destination load balancing, the packet arrives at the core router and the hash value is computed based on the source IP address, destination IP address, and router ID. The PXF has a proprietary algorithm to select a path based on the number of total paths available.

Per-packet load balancing allows data traffic to be evenly distributed in an IP network over multiple equal-cost connections. Per-packet load balancing uses round-robin techniques to select the output path without basing the choice on the packet content.

Multiple Ingress and Multiple Egress Provider Edge Applications

Figure 1-20 shows multiple IGP paths from PE to PE for iBGP paths into the PE2 router. The theoretical load balance is eight IGP paths multiplied by eight iBGP paths for a total of 64 possible unique paths. The Cisco 10000 series router supports eight unique paths. The "Single Ingress and Single Egress Provider Edge Applications" section on page 1-14 describes the path selection for this model.

Figure 1-20 Multiple Ingress and Multiple Egress PE Load Balancing



New Features, Enhancements, and Changes

The following sections describe features that are new, enhanced, or changed for the specified Cisco IOS software releases:

- New Features in Cisco 10000 Series Router Software Configuration Guide - IOS Release 12.2(33)SB, page 1-16

New Features in Cisco 10000 Series Router Software Configuration Guide - IOS Release 12.2(33)SB

In Cisco IOS Release 12.2(33)SB support was added on the Cisco 10000 series router for the following features:

- Unicast Reverse Path Forwarding (uRPF)

  For more information, see Chapter 13, "Unicast Reverse Path Forwarding"
- Any Transport over MPLS (AToM): Tunnel Selection

  For more information, see the "Any Transport over MPLS—Tunnel Selection" section on page 17-47
- L2VPN Interworking: Ethernet/VLAN to ATM AAL5

  For more information, see the "Ethernet/VLAN to ATM AAL5 Interworking" section on page 18-4
- L2VPN Interworking: Ethernet/VLAN to Frame Relay

  For more information, see the "Ethernet/VLAN to Frame Relay Interworking" section on page 18-13
- IPv6 VPN over MPLS (6VPE)

  For more information, see the "IPv6 VPN over MPLS (6VPE)" section on page 4-6
- Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown

  For more information, see the "Remote Ethernet Port Shutdown" section on page 17-25
- NSF / SSO - Any Transport over MPLS (AToM)

  For more information, see the "NSF and SSO—L2VPN" section on page 17-6
- L2VPN Local Switching--HDLC/PPP

  For more information, see the "L2VPN Local Switching—HDLC/PPP" section on page 17-10
- MLP at LNS

  For more information, see the "MLP on LNS" section on page 19-18
- IEEE 802.1Q Tunneling (QinQ) for AToM

  For more information, see the "IEEE 802.1Q Tunneling for AToM—QinQ" section on page 17-22
- IGP Convergence Acceleration

  This feature allows faster failover of IGP routes in load balanced situation.

- Gigabit EtherChannel-Enhancements

  For more information, see Chapter 20, "Configuring Gigabit EtherChannel Features"

- ISG:Flow Control: Flow redirect (PXF scaling)

  For more information, see "Layer 4 Redirect Scaling" section on page 2-4

- VRF-Aware VPDN Tunnels

  This feature places broadband traffic in a VRF based on the VPDN group. This allows more flexible DSL service at the Layer 2 Network Server (LNS).

New Features in Cisco IOS Release 12.2(31)SB5

In Cisco IOS Release 12.2(31)SB5 support was added for the following features:

- Generic Routing Encapsulation (GRE) Tunnel IP Source and Destination VRF Membership

  For more information, see the "GRE Tunnel IP Source and Destination VRF Membership" section on page 24-1.

- Per Session Queuing and Shaping for PPPoE Over VLAN Using RADIUS

  For more information, see the "Shaping PPPoE Over VLAN Sessions Using RADIUS" section in the "Configuring Dynamic Subscriber Services" chapter of the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

New Features in Cisco IOS Release 12.2(31)SB3

In Cisco IOS Release 12.2(31)SB3, support was added on the Cisco 10000 series router for the following features and functionality:

- IS-IS-MIB

  For more information, see the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122sg25/ismibspt.htm

- QoS: MQC Classification, Policing, and Marking on LAC

Note Support for this feature on the PRE3 was introduced in Cisco IOS Release 12.2(31)SB2.

  For detailed information about this feature, see the "Shaping Traffic" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- TCP MSS Adjust

  For more information, see the "Configuring PPPoE over Ethernet and IEEE 802.1Q VLAN" chapter in the *Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide*, located at the following URL:

http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00804d45ca.html

New Features in Cisco IOS Release 12.2(31)SB2

In Cisco IOS Release 12.2(31)SB2, support was added on the Cisco 10000 series router for the following features and functionality:

- ACL - Template ACL/12 Bit ACE

  For more information, see the "Configuring Template ACLs" section on page 22-1.

- Frame Relay - Multilink (MLFR-FRF.16)

  For more information, see the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a0080134a9e.html

- IEEE 802.1Q-in-Q VLAN Tag Termination

  Support was added for the PRE3. For more information, see the following URL:
  http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html

- IP Options Selective Drop

  For more information, see the "Protecting the Router from DoS Attacks" section on page 23-1.

- IPv6 Services: Extended Access Control Lists

  For more information, see the "IPv6 Extended ACLs" section on page 21-4.

- L2TP Domain Screening

  For more information, see the following URL:
  http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00805a0782.html

- L2VPN Interworking — Ethernet to VLAN Interworking

  For more information, see the "Ethernet to VLAN—Bridged Interworking" section on page 18-2.

- MLPPP - Multilink PPP

  Support was added for the PRE3 and the valid multilink interface values on the PRE2 and PRE3 for MLP over Serial and Multi-VC MLP over ATM changed from 1 to 9999 (Release 12.2(28)SB and later) to from 1 to 9999 and 65,536 to 2,147,483,647. For more information, see the "Configuring Multilink Point-to-Point Protocol Connections" section on page 19-1.

- MPLS VPN-VRF Selection based on Source IP Address

  For more information, see the following URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122sz/12214sz/122szvrf.htm

- Multicast VPN Extranet Support

  For more information, see the following URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/extvpnsb.htm

- Multicast VPN Extranet VRF Select

  For more information, see the following URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/sbmexsel.htm

- NSF/SSO (Nonstop Forwarding with Stateful Switchover)

  Support was added for the PRE3. For more information, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s20/fsnsf20s.htm

- QoS - Policing Support for GRE Tunnels

  For more information, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/grepol.htm

- SSO - Multilink Frame Relay

  For more information, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s20/fssso20s.htm

- VRF-Aware VPDN Tunnels

  For more information, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb28/sbvpdnmh.htm

New Features in Cisco IOS Release 12.2(28)SB1

IEEE 802.1Q-in-Q VLAN Tag Termination in the *PPPoE—QinQ Support* feature guide, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html

New Features in Cisco IOS Release 12.2(28)SB

The following features are new on the Cisco 10000 series router in Cisco IOS Release 12.2(28)SB:

- AAA CLI Stop Record Enhancement in the *Per VRF AAA* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a0080518ac1.html

- Any Transport Over MPLS: Frame Relay over MPLS (FRoMPLS) in Chapter 17, "Configuring L2 Virtual Private Networks"

- Cisco 10000 series router 4-Port Channelized T3 Half-Height line card (new line card) in the following guides:

  – *Cisco 10000 Series Router Line Card Configuration Guide*, located at the following URL:

    http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a008071145e.html

  – *Cisco 10000 Series Router Line Card Hardware Installation Guide*, located at the following URL:

    http://www.cisco.com/en/US/products/hw/routers/ps133/products_installation_guide_book09186a00804c9489.html

- Cisco 10000 series 4-Port OC-3/STM-1c ATM line card (long reach optics added to the existing line card) in the *Cisco 10000 Series Router Line Card Hardware Installation Guide, located* at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_installation_guide_book09186a00804c9489.html

- Commands:
  - Changes to **show pxf** command output.
  - New commands (**pos flag s1-byte tx** and **pos flag s1-byte rx-communicate)** for Packet Over SONET and ATM line cards in the *Cisco 10000 Series Router Line Card Configuration Guide, located* at the following URL:

    http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a008071145e.html
  - Changes to the **show running vrf** command in the *MPLS VPN—Show Running VRF* feature guide, located at the following URL:

    http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a00805f236c.html
  - New command for providing policy map information in the *QoS: Enhanced Show Commands for Active Policies* feature guide, located at the following URL:

    http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a0080610cc8.html

- Define Interface Policy-Map AV Pairs AAA in the *Define Interface Policy-Map AV Pairs AAA* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a0080335ed5.html

- Frame Relay PVC Interface Priority Queueing in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- Hierarchical Input Policing in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- IGMPv3 in the “Configuring IGMP Version 3” section in the “Configuring IP Multicast Routing” chapter of “Part 3: IP Multicast” of the *Cisco IOS IP Configuration Guide, Release 12.2*.

- In Service Software Upgrade (ISSU) in the *Cisco IOS In Service Software Upgrade and Enhanced Fast Software Upgrade Process* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a008063c6e7.html

- Intelligent Service Architecture features in the *Intelligent Service Gateway (ISG) Configuration Library*, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a008064ec11.html

- IP SLAs—LSP Health Monitor in the *IP SLAs—LSP Health Monitor* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a0080528450.html

- IPv6 in Chapter 21, "Configuring IP Version 6"
- L2TP Congestion Avoidance in the *L2TP Congestion Avoidance* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a00805f040e.html
- Layer 2 Local Switching in Chapter 17, "Configuring L2 Virtual Private Networks"
- Link Fragmentation Interleave Over Frame Relay (FRF.12) in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Logging to Local Non-Volatile Storage (ATA Disk) in the *Syslog Writing to Flash* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a0080611212.html
- MLP Connections in Chapter 19, "Configuring Multilink Point-to-Point Protocol Connections"
- MLPPP with Link Fragmentation Interleave (LFI) in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- MPLS Carrier Supporting Carrier (also known as MPLS VPN—Carrier Supporting Carrier) in the following feature guides. These guides are located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

  - LDP: MPLS VPN—Carrier Supporting Carrier
  - *BGP: MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution*
- MPLS Embedded Management—LSP Ping/Traceroute and AToM VCCV in the *MPLS Embedded Management—LSP Ping/Traceroute and AToM VCCV* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a008063d009.html
- MPLS Egress Netflow Accounting in the *MPLS Egress Netflow Accounting* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a0080611269.html
- MPLS High Availability Overview in the *MPLS High Availability: Overview* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a00805ad326.html

Note In Cisco IOS Release 12.2(28)SB, the Cisco 10000 series router supports Route Processor Redundancy Plus (RPR+) and Stateful Switchover (SSO). However, for broadband aggregation features the router supports RPR+ only.

- NSF/SSO—MPLS LDP and LDP Graceful Restart in the *NSF/SSO—MPLS LDP and LDP Graceful Restart* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a008029b285.html
- NSF/SSO—MPLS VPN in the *NSF/SSO—MPLS VPN* feature guide, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a00805ad34f.html

- MPLS High Availability: Command Changes in the *MPLS High Availability: Command Changes* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a00805ad151.html

- Cisco Express Forwarding: Command Changes in the *Cisco Express Forwarding: Command Changes* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a008029b100.html

- MPLS—LDP MD5 Global Configuration in the *MPLS—LDP MD5 Global Configuration* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a00805f24da.html

- MPLS VPN—Explicit Null Label Support with BGP IPv4 Label Session in the *MPLS VPN—Explicit Null Label Support with BGP IPv4 Label Session* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

- Load Splitting IP Multicast Traffic—For more information about configuring native multicast load splitting, see the configuration document located at the following URL:

  http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a00805a595a.html

Note You should not configure native multicast load splitting for PE devices running EIBGP as this can result in a loss of traffic.

- Multicast-VPN: Multicast Support for MPLS VPN in the *Multicast VPN—IP Multicast Support for MPLS VPNs* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a008061128c.html

- Nonstop Forwarding with Stateful Switchover (NSF/SSO) in the *Cisco Nonstop Forwarding* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_white_paper09186a00801ce6f5.shtml

- Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services in the *Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a00805f5112.html

- RADIUS Server Load Balancing in the *RADIUS Server Load Balancing* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/ps6566/products_feature_guide09186a008063cffe.html

- Scaling limits for L2TP tunnels in Scaling Enhancements in Cisco IOS Release 12.2(28)SB, page 2-8

- SSO—Multilink PPP (MLP) in the *Stateful Switchover* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_white_paper09186a00801ce6f9.shtml

Note In Cisco IOS Release 12.2(28)SB, the Cisco 10000 series supports Route Processor Redundancy Plus (RPR+) and Stateful Switchover (SSO). However, for broadband aggregation features the Cisco 10000 series supports RPR+ only.

- Template ACLs in Chapter 22, "Configuring Template ACLs"
- Two-Rate Policer (also known as Dual Rate Three Color Policer) in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Cisco IOS Release 12.2(28)SB Upgrade in the *Upgrading to Cisco IOS Release 12.2(28)SB on a Cisco 10000 Series Router*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_upgrade_guides09186a008059adee.html

New Features in Cisco IOS Release 12.3(7)XI7

The following features are new on the Cisco 10000 series router in Cisco IOS Release 12.3(7)XI7:

- Dynamic Subscriber Bandwidth Selection in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- L2TP Domain Screening, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00805a0782.html
- Per Session Queuing and Shaping for PTA in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Support for IP over Q-in-Q (IPoQ-in-Q)—IP packets that are double-tagged for Q-in-Q VLAN tag termination on the subinterface level. For more information, see the *PPPoE—QinQ Support* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html
- VRF-Aware VPDN Tunnels, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a0080509f81.html

New Features in Cisco IOS Release 12.3(7)XI3

The following features are new on the Cisco 10000 series router in Cisco IOS Release 12.3(7)XI3:

- PPPoE Circuit-Tag Processing in the *PPPoE Profiles* feature guide, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00801541b8.html

- QoS: Broadband Aggregation Enhancements - Phase 1 (LAC QoS) in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

New Features in Cisco IOS Release 12.3(7)XI2

The following features are new on the Cisco 10000 series router in Cisco IOS Release 12.3(7)XI2:

- Define Interface Policy-Map AV Pairs AAA in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- Configuring atm pxf queuing, page 2-16(scaling enhancements)
- Dynamic ATM VP and VC Configuration Modification in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- Local Template-Based ATM PVC Provisioning, page 8-2
- MQC Policy Map Support on Configured VC Range in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- RADIUS Attribute 31: PPPoX Calling Station ID, page 16-13
- Scaling Enhancements in Cisco IOS Release 12.3(7)XI2, page 2-7
- Shaped UBR PVCs in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

New Features in Cisco IOS Release 12.3(7)XI1

While some of the following features are supported on other releases on the Cisco 10000 series router, these features are new in Cisco IOS Release 12.3(7)XI1:

- 3-Color Policer in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- 3-Level Hierarchical QoS Policies in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN, page 4-1
- Class-based Weighted Fair Queueing in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Extended NAS-Port-Type and NAS-Port Support, page 16-6
- Half-Duplex VRF, page 4-20
- Hierarchical Shaping in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- IEEE 802.1Q-in-Q VLAN Tag Termination in the *PPPoE—QinQ Support* feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html
- Interface Oversubscription in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- IP Receive ACLs, page 12-1
- Configuring IP Unnumbered on IEEE 802.1Q VLANs, page 7-1
- Configuring Local AAA Server, User Database—Domain to VRF, page 11-1
- MPLS QoS in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- MPLS Traffic Engineering—Diffserv Aware in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- MR-APS in Configuring Automatic Protection Switching, page 14-1
- Percent-Based Policing in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Per DSCP Weighted Random Early Detection in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html
- Per Precedence Weighted Random Early Detection Statistics in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- PPPoE over Q-in-Q (PPPoEoQ-in-Q)—PPPoE packets that are double-tagged for Q-in-Q VLAN tag termination on the subinterface level. For more information, see the PPPoE—QinQ Support feature guide, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801f0f4a.html

- RADIUS Packet of Disconnect, page 16-17
- Scaling Enhancements in Cisco IOS Release 12.3(7)XI1, page 2-6
- Time-Based ACLs, page 12-4
- Variable Bit Rate Non-Real Time Oversubscription, page 8-14
- VC Weighting in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

- Weighted Random Early Detection with Queue Limit in the *Cisco 10000 Series Router Quality of Service Configuration Guide*, located at the following URL:

  http://www.cisco.com/en/US/products/hw/routers/ps133/products_configuration_guide_book09186a00805b9497.html

CHAPTER 2

Scalability and Performance

The infrastructure of the service provider must be capable of supporting the services the enterprise customer or Internet service provider (ISP) wants to offer its subscribers. It must also be able to scale to an expanding subscriber base. You can configure the Cisco 10000 series router for high scalability.

This chapter discusses the following topics:

Line Card VC Limitations

The Cisco 10000 series router supports four ATM service categories for virtual circuits (VCs):

- Constant Bit Rate (CBR)
- Variable Bit Rate-nonreal-time (VBR-nrt)
- Unspecified Bit Rate (UBR) with a peak cell rate (PCR), referred to as shaped UBR
- UBR without a PCR, referred to as unshaped UBR

The segmentation and reassembly (SAR) mechanism configures priority and additional traffic management parameters for the various ATM service categories. Table 2-1 lists the priority levels the SAR sets for the service categories.

Table 2-1 ATM Service Categories

Parameter	CBR	VBR-rt	VBR-nrt	Shaped UBR	Unshaped UBR
Priority	0	1	2	3	None

The number of SAR priority levels and the service categories supported at each priority level vary from line card to line card. For example, the 1-port OC-12/STM-1 line card supports the four levels of priority and the service categories listed in Table 2-2, but the 4-port OC-3 line card supports only two levels of priority and the service categories listed in the table.

The ATM line cards support a maximum number of VCs per priority. That VC limit depends on the VC limit of the SAR (SAR limit) and the number of priority levels configured. Table 2-2 describes how to determine the VC limit per priority level per port for the specified line cards.

Table 2-2 Maximum Number of VCs per Priority

ATM Line Card	SAR Priority Levels	VC Rate	Maximum Number of VCs per Priority
1-Port OC-12/ STM-1	0 = CBR VCs 1 = VBR-rt VCs 2 = VBR-nrt VCs 3 = UBR VCs	Full line rate	SAR limit / 2 / number of priority levels 4 priority system: 65,536 / 2 / 4 = 8192 VCs per priority level
		Half line rate and below	SAR limit / number of priority levels 4 priority system: 65,536 / 4 = 16,384 VCs per priority level
4-Port OC-3	0 = CBR, VBR-nrt VCs 1 = UBR VCs	Half line rate and below	SAR limit / number of PHYs / number of priority levels 2 priority system: 65,536 / 4 / 2 = 8192 VCs per priority level per port
8-Port E3/DS3	0 = CBR VCs 0 = VBR-nrt VCs 1 = UBR VCs	Half line rate and below	SAR limit / number of PHYs / number of priority levels 2 priority system: 65,536 / 8 / 2 = 4096 VCs per priority level per port

Configuring more channels or VCs than there are available priority locations can cause random channels or VCs to get stuck in the SAR. This occurs when an active channel tries to reschedule itself, but no priority locations are available. Therefore, the channel cannot find a place to reschedule itself, which results in a lost event for the channel, and the channel becomes stuck in the SAR.

On the PRE2, when a VC becomes stuck in the SAR, the PRE2 scheduler stops forwarding traffic on only the VC that is stuck in the SAR; the other VCs still carry traffic. On the PRE3, the PRE3 scheduler stops forwarding traffic on all the VCs configured on that ATM line card.

For example, suppose a 1-port OC-12 line card at full line rate is configured for four levels of priority and a 4-port OC-3 line card at half line rate is configured for two levels of priority. By calculating the maximum number of VCs as described in Table 2-2, you can configure 8192 VCs per priority level for the 1-port OC-12 and 8192 VCs per priority level per port for the 4-port OC-3—a total of 16,384 VCs per priority level per port. If the number of VCs you configure exceeds the VC limit, the VCs get stuck in the SAR.

Limitations and Restrictions

The Cisco 10000 series router has the following limitations and restrictions for scalability and performance:

- When Layer 4 Redirect (L4R) service is applied without Port Bundle Host Key (PBHK) service, the translations are all done in the PXF, except for those translations that encounter a collision condition. A collision occurs when a subscriber has two simultaneous TCP connections whose source ports have the same Modulo 64 result.

  For example, the subscriber has an active TCP connection on source port 1026, and while this connection is still alive the subscriber starts another TCP connection on source port 1090. A collision is created because the Modulo 64 result for both the source ports (1024 and 1090) is 2. In this example, L4R translation for the first traffic stream is done in the PXF and for the second TCP stream the packets are sent to the route processor (RP) where the L4R translation is done. This seperation prevents collisions.

- When the PBHK service is applied with L4R service, certain restrictions apply:
  - When the destination IP in any one of the access control entries of the PBHK ACL matches the redirected server IP address, then both L4R and PBHK translations are done in the RP.
  - When the destination IP address in the access control entries of the PBHK ACL does not match the redirect server IP address, then L4R translations are done in the PXF, and the packets that match the PBHK ACL are translated in the RP.

  For configuration examples, see the "Layer 4 Redirect Scaling" section on page 2-4.

- Certain restrictions apply on L4R translations for IP subnet sessions. If two subscribers send TCP traffic using the same source port, then L4R translation for the common port is done in the RP. However, if a group of IP subscribers in an IP subnet session send traffic on different source ports then L4R translations for all the subscribers are done in the PXF.
- For permanent L4R service, you can scale up to the number of sessions listed in Table 2-3. Scaling beyond these sessions can lead to an increase in CPU usage that is beyond the recommended limits.

Table 2-3 Scaling Limit of L4R Sessions

Cisco IOS Release	PRE2	PRE3	PRE4
12.2(31)SB	4000	4000	—
12.2(33)SB	4000	16000	16000
12.2(34)SB	4000	16000	16000

- You can apply access control lists (ACLs) to virtual access interfaces (VAIs) by configuring them under virtual template interfaces. You can also configure ACLs by using RADIUS attribute 11 or 242. Prior to Cisco IOS Release 12.2(28)SB, when you used attribute 242, a maximum of 30,000 sessions could have ACLs; this restriction was removed in release 12.2(28)SB and subsequent releases.
- For PRE2, the Cisco 10000 series router supports mini-ACLs (eight or fewer access control entries) and turbo ACLs (more than eight access control entries) for non-SSG interfaces. The limit for mini-ACLs is 32,000. The limit for turbo ACLs depends on the complexity of the defined ACLs. For PRE3, the Cisco 10000 series router does not use mini-ACLs.
- For SSG (RADIUS) configurations on PRE2, the following limitations apply:

– For Cisco IOS Release 12.3(7)XI, ACLs defined through SSG configuration (RADIUS) are restricted to mini-ACLs only. Turbo ACLs cannot be used in combination with SSG and RADIUS. If you apply a Turbo ACL to an SSG session, the following syslog error is generated: "%C10K_ACLS-3-SSG_TURBO_ACL: acl is a Turbo ACL and cannot be used for SSG."

Note If a mini-ACL is on the verge of becoming a turbo ACL (that is, the ACL contains eight access control entries), SSG redirection can cause the mini-ACL to become a turbo ACL. For Cisco IOS Release 12.3(7)XI, this change would also cause a syslog error to be generated as follows: "%C10K_ACLS-3-SSG_ACL_ERR: acl is miniACL but cannot have another punt rule added."

- The Cisco 10000 series router supports a maximum of 2,000 authentication, authorization, and accounting (AAA) method lists. If you configure more than 2,000 AAA method lists by using the **aaa authentication ppp** or **aaa authorization network** command, traceback messages appear on the console.
- To avoid CPU overload and router instability, use the **logging rate-limit** command to limit the rate that the Cisco 10000 series router logs system messages. For more information, see the **logging rate-limit** command in the *Cisco IOS Configuration Fundamentals and Network Management Command Reference, Release 12.3*, located at the following URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_command_reference_book09186a008017d0a2.html
- The Cisco 10000 series router high-speed interfaces work efficiently to spread traffic flows equally over the queues. However, using single traffic streams in a laboratory environment might result in less-than-expected performance. To ensure accurate test results, test the throughput of the Gigabit Ethernet, OC-48 POS, or ATM uplink with multiple source or destination addresses. To determine if traffic is being properly distributed, use the **show pxf cpu queue** command.
- The Cisco 10000 series router supports a configuration file of up to 16 megabytes.
- If you configure create on demand PVCs (individual and within a range) and PPP sessions, RP CPU utilization can be extremely high when bringing up and tearing down sessions and PVCs. This usage is a concern only when the configuration contains approximately 30,000 PPP sessions, and additional services are enabled (such as DBS, ACLs, and service policies).

  To reduce the RP CPU usage for PPPoA sessions, reduce the number of configured PVCs in a single subinterface. To reduce the RP CPU usage for PPPoEoA sessions, use call admission control (**call admission limit** command).

Scaling Enhancements in Cisco IOS Release 12.2(33)SB

Cisco IOS Release 12.2(33)SB provides increased scalability for the Layer 4 Redirect feature.

Layer 4 Redirect Scaling

The Layer 4 Redirect feature allows redirection of users' TCP or UDP traffic to a server to control and increase performance. In Cisco IOS Release12.2(33)SB, the ISG L4R feature is implemented in the PXF. This design increases the number of redirects to provide higher scalability and performance. This enhancement is a scalable solution for portals and self-provisioning and is supported on PRE3 and PRE4 only. On a PRE2 L4R translations are done in the RP.

PBHK translations are always done in the RP. The L4R feature is scalable when applied alone; however, certain scalability restrictions apply when it is used with PBHK. See also the "Limitations and Restrictions" section on page 2-3.

In Example 2-1, when the destination IP used in the PBHK ACL (162) matches the redirected server IP address, L4R translations are done in the RP.

Example 2-1 L4R Translations in the Route Processor

```
class-map type traffic match-any class-l4r
match access-group input 152

policy-map type service ser-l4r
class type traffic class-l4r
redirect to ip 200.0.0.2

ip portbundle
match access-list 162
source loopback 1

access-list 152 deny tcp any host 200.0.0.2
access-list 152 permit tcp any any

access-list 162 permit tcp any host 200.0.0.2
```

In Example 2-2, when the destination IP used in the PBHK ACL (162) is not the same as the redirected server IP address, L4R translations are done in the PXF.

Example 2-2 L4R Translations in PXF

```
class-map type traffic match-any class-l4r
match access-group input 152

policy-map type service ser-l4r
class type traffic class-l4r
redirect to ip 210.0.0.2

ip portbundle
match access-list 162
source loopback 1

access-list 152 deny tcp any host 200.0.0.2
access-list 152 permit tcp any any

access-list 162 permit tcp any host 200.0.0.2
```

For more information on configuring L4R, see the "Redirecting Subscriber Traffic Using ISG Layer 4 Redirect" chapter in the *Cisco IOS Intelligent Service Gateway Configuration Guide, Release 12.2 SB* at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_configuration_guide_chapter09186a0080630d65.html#wp1048970

For more information on configuring PBHK, see the "Configuring ISG Port-Bundle Host Key" chapter in the *Cisco IOS Intelligent Service Gateway Configuration Guide, Release 12.2 SB* at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_configuration_guide_chapter09186a0080630d6c.html

Scaling Enhancements in Cisco IOS Release 12.3(7)XI1

Cisco IOS Release 12.3(7)XI1 provides increased limits with FIB scaling, policy-map scaling, and queue scaling.

FIB Scaling

The FIB is a routing table that is used to look up the next hop route for the destination IP address and the reverse path forwarding (RPF) route using the source IP address. The FIB Scaling feature implements the following changes:

- Up to 1 million routes in the global FIB table are supported without MPLS VPN configuration.
- Total number of virtual routing and forwarding instances (VRFs) supported is 4095.
  - Up to 100 routes per VRF with 4095 VRFs configured.
  - Up to 70 routes per VRF with 4095 VRFs configured, plus 200,000 global BGP routes.
  - Up to 600 routes per VRF with 1000 or fewer VRFs configured.

Policy-Map Scaling

The Policy-Map Scaling feature increases the system-wide number of quality of service (QoS) policy maps that you can configure. Depending on the complexity of your configuration, the Cisco 10000 series router supports up to 4096 policy maps. In complex configurations the maximum number of policy maps can be as small as a few hundred. Additionally, when you use percent-based policing in a service policy, the system may convert a single customer-configured service to multiple service policies (which count against the 4096 limit). The system uses one such service policy for each different speed interface that uses a service policy with percent-based policing

Each **policy-map** command counts as one policy map and applying the same policy map on different speed interfaces also counts as an extra policy map. The **policy-map** command syntax is unchanged. The maximum number of classes that you can configure in a policy is 127.

Queue Scaling

The Queue Scaling feature increases the total number of queues that VTMS supports to 131,072. Of the total number, 254 queues are available for high speed interfaces, and 130,816 queues are available for low speed interfaces. This increase allows the support of the 31,500 priority queues (of 131,072 total queues) on 31,500 sessions or interfaces.

Each interface includes a class-default queue and a system queue. If you attach an output policy map with 1 priority queue and 1 class-based weighted fair queue (PQ/CBWFQ) to each of the 31,500 interfaces, the number of priority queues is 31,500 and the total number of queues is 31,500 x 4, or 126,000 queues.

The maximum number of queues per link remains at 32, of which 29 are user-configurable because there is 1 class-default queue, 1 system queue, and 1 reserved queue.

To support 131,072 queues, the queue limits range has changed. For high-speed interfaces (an interface that has a speed greater than 622 Mbps), the queue limit range is 128 to 65,536. For low-speed interfaces the queue limit range is 8 to 4,096. Because the total number of packet buffers for queue limits is 4,194,304, the average queue depth is less than or equal to 32 per queue with 131,072 queues configured.

On low-speed interfaces, the default queue size is 8 for all QoS CBWFQ queues, with the exception of WRED queues. The default queue size for WRED queues is 32.

The class-default queue size on low-speed interfaces has changed from 32 to 8. If the traffic is too bursty and packets drop, you can use the **queue-limit** command to increase the class-default queue size.

If you change the queue size for 131,072 queues while traffic is running, the queue size for a few queues might not be changed if packets were in the queues. An "out of resource" message can also appear. Use the **queue-limit** command to modify the queue size for those queues that were not changed.

The queue limits packet buffers can become fragmented after the queue sizes on 131,072 queues has been changed a few times. The system might indicate that there are not enough resources to increase queue size, even though there are enough free packet buffers. Removing and reapplying the policy map on the interfaces solves this problem.

Use the **show pxf cpu queue summary** command to see the number of packet buffers, packet buffers being recycled, and free packet buffers.

Scaling Enhancements in Cisco IOS Release 12.3(7)XI2

Cisco IOS Release 12.3(7)XI2 provides increased limits with queue scaling and VC scaling.

Queue Scaling

At least two queues are allocated for every interface or subinterface for which separate queues are created. The first queue is the default queue for normal traffic, and the second queue, known as the system queue, is used for a small amount of router-generated traffic that bypasses the normal drop mechanisms. For 32,000 VCs, this setup would require the allocation of a minimum of 64,000 queues. While Cisco IOS Release 12.3(7)XI1 adds support for up to 128,000 queues, a more effective use of these limited resources is realized by having the subinterfaces on a given main interface share the single system queue of the main interface.

In Cisco IOS Release 12.3(7)XI2, the subinterfaces on a given main interface share the single system queue of the main interface, which allows for 32,000 subinterfaces with a three-queue model that supports assured forwarding (AF) queues and expedited forwarding (EF) queues, in addition to the default best effort (BE) queues. Because a system queue does not exist for every subinterface, this setup frees up queues for a 4-queue model.

VC Scaling

When configured for hierarchical shaping, ATM line cards support the following number of VCs:

- E3/DS3 line card supports a maximum of 4,096 VCs
- OC-12 ATM line card supports a maximum of 16,384 VCs (previously 14,436)
- OC-3 ATM line card supports a maximum of 8,191 VCs

Scaling Enhancements in Cisco IOS Release 12.2(28)SB

In Cisco IOS Release 12.2(28)SB, up to 16,384 L2TP tunnels are supported. Because of a limit on the number of VPDN groups supported, it is not possible to configure 16,384 tunnel definitions using the CLI. Configure the remaining tunnel definitions using RADIUS.

Configuring the Cisco 10000 Series Router for High Scalability

To ensure high scalability on the Cisco 10000 series router, perform the following configuration tasks:

Configuring Parameters for RADIUS Authentication

If your network uses a RADIUS server for authentication, set the small, middle, and big buffers by using the **buffers** command. Table 2-4 lists the buffer sizes to configure (and see Example 2-3).

Table 2-4 Buffer Sizes for RADIUS Authentication

Buffer	Size
Small	15000
Middle	12000
Big	8000

Example 2-3 Configuring Buffer Sizes

```
Router(config)# buffers small perm 15000
Router(config)# buffers mid perm 12000
Router(config)# buffers big perm 8000
```

Typically, if the RADIUS server is only a few hops away from the router, we recommend that you configure the RADIUS server retransmit and timeout rates by using the **radius-server** command. Table 2-5 lists the recommended settings (and see Example 2-4).

Table 2-5 RADIUS Server Parameters

Parameter	Value
RADIUS Server Retransmit Rate	5
RADIUS Server Timeout Rate	15

Example 2-4 Configuring RADIUS Server Parameters

```
Router(config)# radius-server retransmit 5
Router(config)# radius-server timeout 15
```

Configuring L2TP Tunnel Settings

Configure an L2TP tunnel password using Cisco IOS Release 12.2(4)BZ1 or later. We recommend that you configure the L2TP tunnel parameters listed in Table 2-6 (and see Example 2-5, Example 2-6, and Example 2-7).

Table 2-6 L2TP Tunnel Settings

Parameter	Setting
No Session Timeout	30
L2TP Tunnel Receive Window	100
L2TP Tunnel Retransmit Timeout	2 (minimum) 8 (maximum)

Note The No Session Timeout parameter indicates the length of time a tunnel persists when there are no sessions in the tunnel.

Example 2-5 Configuring an L2TP Tunnel Password

```
Router(config)# vpdn-group tunnel1
Router(config-if)# l2tp tunnel password 7
```

Example 2-6 Configuring the No Session Timeout Parameter

```
Router(config)# vpdn-group tunnel1
Router(config-if)# l2tp tunnel nosession-timeout 30
```

Example 2-7 Configuring the L2TP Tunnel Receive-Window and Retransmit Timeout Parameters

```
Router(config)# vpdn-group tunnel1
Router(config-if)# l2tp tunnel receive-window 100
Router(config-if)# l2tp tunnel retransmit timeout min 2
Router(config-if)# l2tp tunnel retransmit timeout max 8
```

VPDN Group Session Limiting

Before the introduction of the VPDN Group Session Limiting feature introduced in Cisco IOS software release 12.2(1)DX, you could only globally limit the number of VPDN sessions on a router with limits applied equally to all VPDN groups. Using the VPDN Group Session Limiting feature, you can limit the number of VPDN sessions allowed per VPDN group. For more information, see the VPDN Group Session Limiting feature documentation, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5012/products_feature_guide09186a0080087ef2.html

Configuring the PPP Authentication Timeout

To keep the L2TP network server (LNS) from timing out a PPP authentication process, set the PPP Timeout parameter to 100, using the **ppp timeout authentication** command (Example 2-8).

Example 2-8 Configuring the PPP Authentication Timeout

```
Router(config)# interface Virtual-Template1
Router(config-if)# ppp timeout authentication 100
```

Disabling Cisco Discovery Protocol

To maximize scalability, do not enable the Cisco Discovery Protocol (CDP).

Note CDP is disabled by default.

Disabling Gratuitous ARP Requests

To maximize the performance of the router, disable gratuitous ARP requests, using the **no ip gratuitous-arp** command (Example 2-9).

Example 2-9 Disabling Gratuitous ARP Requests

```
Router(config)# no ip gratuitous-arp
```

Configuring a Virtual Template Without Interface-Specific Commands

If you configure a virtual template interface with interface-specific commands, the Cisco 10000 series router does not achieve the highest possible scaling. To verify that the router does not have interface-specific commands within the virtual template interface configuration, use the **test virtual-template** <*number*> **subinterface** command.

Including interface-specific commands in a virtual template can limit PPP session scaling. Table 2-7 lists the interface-specific commands that prevent the Cisco 10000 series router from attaining the highest possible PPP session scaling.

Table 2-7 Interface-Specific Commands That Prevent PPP Scaling

Command	Function
access-expression	Builds a bridge Boolean access expression.
asp	Asynchronous Port (ASP) subcommands.
autodetect	Autodetects encapsulations on serial interfaces.
bridge-group	Transparent bridging interface parameters.
bsc	Binary Synchronous Communications (BSC) interface subcommands.
bstun	Block Serial Tunnel (BSTUN) interface subcommands.
carrier-delay	Specifies delay for interface transitions.
cdp	Cisco Discovery Protocol (CDP) interface subcommands.
clock	Configures the serial interface clock.
compress	Sets the serial interface for compression.
custom-queue-list	Assigns a custom queue list to an interface.
diffserv	Differentiated Services (diffserv) for provisioning.
down-when-looped	Forces a looped serial interface down.
encapsulation	Sets the encapsulation type for an interface.
fair-queue	Enables fair queuing on an interface.
full-duplex	Configures full-duplex operational mode.
h323-gateway	Configures the H.323 Gateway.
half-duplex	Configures half-duplex and related commands.
help	Provides a description of the interactive help system.
hold-queue	Sets the hold queue depth.
lan-name	Specifies a name for the LAN that is attached to the interface.
lapb	X.25 Level 2 parameters (Link Access Procedure, Balanced).

Table 2-7 Interface-Specific Commands That Prevent PPP Scaling (continued)

Command	Function
load-interval	Specifies the interval for load calculation for an interface.
locaddr-priority	Assigns a priority group.
logging	Configures logging for an interface.
loopback	Configures the internal loopback on an interface.
mac-address	Manually sets the MAC address for an interface.
max-reserved-bandwidth	Specifies the maximum reservable bandwidth on an interface.
mpoa	Multiprotocol over ATM (MPOA) interface configuration commands.
multilink	Configures multilink parameters.
multilink-group	Puts the interface in a multilink bundle.
netbios	Defines Network Basic Input/Output System (NetBIOS) access list or enables name-caching.
ntp	Configures the Network Time Protocol (NTP).
priority-group	Assigns a priority group to an interface.
qos pre-classify	Enables quality of service (QoS) preclassification.
random-detect	Enables weighted random early detection (WRED) on an interface.
roles	Specifies roles (by entering roles mode).
sap-priority	Assigns a priority group.
sdlc	Configures Synchronous Data Link Control (SDLC) to Logical Link Control type 2 (LLC2) translation.
serial	Serial interface commands.
snmp	Modifies Simple Network Management Protocol (SNMP) interface parameters.
source	Gets the configuration from another source.
stun	Serial Tunnel (STUN) interface subcommands.
transmit-interface	Assigns a transmit interface to a receive-only interface.
trunk-group	Configures an interface to be in a trunk group.
tx-ring-limit	Limits the number of particles or packets that can be used on a transmission ring on an interface.

In Example 2-10, the output of the **test virtual-template** <*number*> **subinterface** command indicates that the interface-specific command **carrier-delay** is set.

Example 2-10 Verifying Interface-Specific Commands in the Virtual Template

```
Router(config)# test virtual-template 11 subinterface

Subinterfaces cannot be created using Virtual-Template11
Interface specific commands:
carrier-delay 45
```

Monitoring PPP Sessions Using the SNMP Management Tools

To prevent the virtual-access subinterfaces from being registered with the SNMP functionality of the router and using memory, do not use the router's SNMP management tools to monitor PPP sessions. Use the **no virtual-template snmp** command to disable the SNMP management tools (Example 2-11).

Example 2-11 Preventing SNMP Registration of Virtual-Access Subinterfaces

```
Router(config)# no virtual-template snmp
```

SNMP Process and High CPU Utilization

Network management applications retrieve information from devices by using SNMP. If a user application polls the SNMP MIBs while the router is updating its routing table, the SNMP engine process can cause CPU HOG messages to appear and sessions and tunnels to go down until the process releases the CPU.

For information about how to avoid high CPU utilization by an SNMP process, see the *IP Simple Network Management Protocol (SNMP) Causes High CPU Utilization* Tech Note, located at the following URL:

http://www.cisco.com/warp/public/477/SNMP/ipsnmphighcpu.shtml#polling

CISCO-ATM-PVCTRAP-EXTN-MIB

The Cisco 10000 series router does not support the CISCO-ATM-PVCTRAP-EXTN-MIB for large numbers of permanent virtual circuits (for example, 32,000 PVCs). To exclude the Cisco-ATM-PVCTRAP-EXTN-MIB from the Simple Network Management Protocol (SNMP) view and enhance scalability, configure the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server view` *view-name oid-tree* **included**	Creates or updates a view entry. The *view-name* argument is a label for the view record that you are updating or creating. The name is used to reference the record. The *oid-tree* argument is the object identifier of the ASN.1 subtree to be included from the view. Specify a valid oid-tree from where you want to poll the information. The **included** argument configures the OID (and subtree OIDs) specified in the *oid-tree* argument to be included in the SNMP view.
Step 2	`Router(config)# snmp-server view` *view-name* **ciscoAtmPvcTrapExtnMIB excluded**	Configures the CISCO-ATM-PVCTAP-EXTN-MIB OID (and subtree OIDs) to be explicitly excluded from the SNMP view. You must specify the oid-tree as shown in the command line. The *view-name* argument must match the *view-name* you specified in step 1.
Step 3	`Router(config)# snmp-server community` *string* [**view** *view-name*] [**ro** | **rw**] [access-list-number]	Sets up the community access string to permit access to SNMP. The *string* argument is a community string that acts like a password and permits access to the SNMP protocol. The *view-name* argument must match the *view-name* you specified in step 1.

Example 2-12 shows how to create or modify the SNMP view named *myview* to include the information polled from the Internet oid-tree and to exclude the CISCO-ATM-PVCTRAP-EXTN-MIB oid-tree. The community access string named private is set up and access to SNMP is read-only (**ro**) access.

Example 2-12 Excluding CISCO-ATM-PVCTRAP-EXTN-MIB from the SNMP View

```
Router(config)# snmp-server view myview internet included
Router(config)# snmp-server view myview ciscoAtmPvcTrapExtnMIB excluded
Router(config)# snmp-server community private view myview ro
```

For more information about the **snmp-server view** and **snmp-server community** commands, see the *Cisco IOS Configuration Fundamentals and Network Management Command Reference, Release 12.3*, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_command_reference_book09186a008017d0a2.html

Configuring the Trunk Interface Input Hold Queue

To ensure high scalability, set the trunk interface input hold queue to a high value (Example 2-13).

Note The default value for the OC-12 ATM line card trunk interface input hold queue is 27230. Cisco laboratory tests have shown this setting to result in the highest scalability for the OC-12 ATM line card. We recommend that you not change the default setting.

Example 2-13 Setting the Trunk Interface Input Hold Queue

```
Router(config)# interface gig1/0/0
Router(config-if)# hold-queue 4096 in
```

Configuring no atm pxf queuing

Note We do not recommend using this mode for QoS-sensitive deployments.

Configuring the **no atm pxf queuing** command on each port of the Cisco 10000 series router enables the router to support a high number of VCs. PPPoA supports one session per VC and requires that you enable **no atm pxf queuing** to support 32,000 PPPoA sessions. Enabling **no atm pxf queuing** is not required for L2TP, and might not be required for PPPoE, because you can have 32,000 sessions on a single VC.

The Cisco 10000 series router supports three ATM traffic classes when you configure **no atm pxf queuing**: unshaped UBR (no PCR is specified), shaped UBR (PCR is specified), and VBR-nrt. To configure an unspecified bit rate (UBR) quality of service (QoS) and specify the output peak cell rate (PCR), use the **ubr** command in the appropriate configuration mode. In ATM VC configuration mode, the syntax is:

```
Router(config-if-atm-vc)# ubr output-pcr
```

If you do not specify a PCR, unshaped UBR is configured.

To configure the variable bit rate-nonreal-time (VBR-nrt) QoS, use the **vbr-nrt** command in the appropriate configuration mode and specify the output PCR, output sustainable cell rate (SCR), and the output maximum burst cell size (MBS) for a VC class. Note that if the PCR and SCR values are equal, the MBS value is 1.

```
output-pcr output-scr output-mbs
```

Note Before you configure VCs on an interface, configure the **atm pxf queuing** mode for the port (**atm pxf queuing** or **no atm pxf queuing**). After you configure the mode, then configure the VCs. Do not change the mode while VCs are configured on the interface. If you need to change the mode, delete the VCs first and then change the mode. Changing the mode while VCs are configured can produce undesired results, and the change will not take effect until the next router reload.

Configuring atm pxf queuing

The Cisco 10000 series router supports two ATM traffic classes when you configure **atm pxf queuing**: unshaped UBR and VBR-nrt. When you specify an output PCR for an unshaped UBR class, the Cisco 10000 series router accepts the PCR. However, the router does not use the PCR value and it does not notify you of this omission.

For information about configuring the traffic classes, see the "Configuring no atm pxf queuing" section on page 2-15.

Note Before you configure VCs on an interface, configure the **atm pxf queuing** mode for the port (**atm pxf queuing** or **no atm pxf queuing**). After you configure the mode, then configure the VCs. Do not change the mode while VCs are configured on the interface. If you need to change the mode, delete the VCs first and then change the mode. Changing the mode while VCs are configured can produce undesired results.

Table 2-8 lists the number of active VCs the ATM line cards support in **atm pxf queuing** mode for Cisco IOS Release 12.3(7)XI2 or later releases.

Table 2-8 Active VCs on ATM Line Cards

Line Card	Maximum VCs per Port	Maximum VCs per Module	No. VBR, CBR, Shaped UBR VCs
E3/DS3	4,096	32,768[1]	28,672[2]
OC-3	8,191	32,764[3]	28,672[4]
OC-12	16,384 (previously 14,436)	16,384	16,384

1. For 32,768 VCs per module, 4096 of them must be unshaped UBR VCs.
2. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.
3. For 32,764 VCs per module, 4096 of them must be unshaped UBR VCs.
4. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.

You can configure the maximum number of VCs across the ports in any fashion, provided that you do not exceed the per-port maximum.

Although the maximum number of VBR, CBR, and shaped UBR VCs per E3/DS3 and OC-3 ATM line card is 28,672 VCs, the router supports a maximum of 22,204 VBR, CBR, and shaped UBR VCs per line card that you can place within virtual path (VP) tunnels. If you attempt to bring up more than 22,204 VCs in a configuration that includes VP tunnels and VCs (hierarchical traffic shaping configuration), the VCs might not assign traffic correctly or the VCs might not come up at all. Be sure to limit the number of configured VBR, CBR, and shaped UBR VCs on an ATM card to less than 22,204 VCs if you place the VCs in VP tunnels.

For the OC-12 ATM line card, the router supports 16,384 VCs in VP tunnels.

Configuring keepalive

The **keepalive** command sets the keepalive timer for a specific interface. To ensure proper scaling and to minimize CPU utilization, set the timer for 30 seconds or longer (Example 2-14). The default value is 10 seconds.

Example 2-14 Configuring keepalive for a Virtual Template Interface

```
interface Virtual-Template1
 ip unnumbered Loopback1
 keepalive 30
 no peer default ip address
 ppp authentication pap
```

Enhancing Scalability of Per-User Configurations

To enhance scalability of per-user configurations without changing the router configuration, use the **ip:vrf-id** and **ip:ip-unnumbered** RADIUS attributes. These per-user vendor specific attributes (VSAs) are used to map sessions to VRFs and IP unnumbered interfaces. The VSAs apply to virtual access subinterfaces and are processed during PPP authorization.

In releases earlier than Cisco IOS Release 12.2(16)BX1, the **lcp:interface-config** RADIUS attribute is used to map sessions to VRFs. This per-user VSA applies to any type of interface configuration, including virtual access interfaces. Valid values of this VSA are essentially any valid Cisco IOS interface command; however, not all Cisco IOS commands are supported on virtual access subinterfaces. To accommodate the requirements of the **lcp:interface-config** VSA, the per-user authorization process forces the Cisco 10000 series router to create full virtual access interfaces, which consume more memory and are less scalable.

In Cisco IOS Release 12.2(16)BX1 and later releases, the **ip:vrf-id** attribute is used to map sessions to VRFs. Any profile that uses the **ip:vrf-id** VSA must also use the **ip:ip-unnumbered** VSA to install IP configurations on the virtual access interface that is to be created. PPP that is used on a virtual access interface to be created requires the **ip:ip-unnumbered** VSA. An Internet Protocol Control Protocol (IPCP) session is not established if IP is not configured on the interface. You must configure either the **ip address** command or the **ip unnumbered** command on the interface so that these configurations are present on the virtual access interface that is to be created. However, specifying the **ip address** and **ip unnumbered** commands on a virtual template interface is not required because any pre-existing IP configurations are removed when the **ip:ip-vrf** VSA is installed on the virtual access interface. Therefore, any profile that uses the **ip:vrf-id** VSA must also use the **ip:ip-unnumbered** VSA to install IP configurations on the virtual access interface that is to be created.

These per-user VSAs can be applied to virtual access subinterfaces; therefore, the per-user authorization process does not require the creation of full virtual access interfaces, which improves scalability.

Setting VRF and IP Unnumbered Interface Configurations in User Profiles

Although the Cisco 10000 series router continues to support the **lcp:interface-config** VSA, the **ip:vrf-id** and **ip:ip-unnumbered** VSAs provide another way to set the VRF and IP unnumbered interface configurations in user profiles. The **ip:vrf-id** and **ip:ip-unnumbered** VSAs have the following syntax:

```
Cisco:Cisco-AVpair = "ip:vrf-id=vrf-name"
Cisco:Cisco-AVpair = "ip:ip-unnumbered=interface-name"
```

You should specify only one **ip:vrf-id** and one **ip:ip-unnumbered** value in a user profile. However, if the profile configuration includes multiple values, the Cisco 10000 series router applies the value of the last VSA received, and creates a virtual access subinterface. If the profile includes the **lcp:interface-config** VSA, the router always applies the value of the **lcp:interface-config** VSA, and creates a full virtual access interface.

In Cisco IOS Release 12.2(15)BX, when you specify a VRF in a user profile, but do not configure the VRF on the Cisco 10000 series router, the router accepts the profile. However, in Cisco IOS Release 12.2(16)BX1 and later releases, the router rejects the profile.

Setting VRF and IP Unnumbered Interface Configuration in a Virtual Interface Template

You can specify one VSA value in the user profile on RADIUS and another value locally in the virtual template interface. The Cisco 10000 series router clones the template and then applies the values configured in the profiles it receives from RADIUS, resulting in the removal of any IP configurations when the router applies the profile values.

Redefining User Profiles to Use the ip:vrf-id and ip:ip-unnumbered VSAs

The requirement of a full virtual access interface when using the **lcp:interface-config** VSA in user profiles can result in scalability issues such as increased memory consumption. This situation is especially true when the Cisco 10000 series router attempts to apply a large number of per-user profiles that include the **lcp:interface-config** VSA. Therefore, when updating your user profiles, we recommend that you redefine the **lcp:interface-config** VSA to the scalable **ip:vrf-id** and **ip:ip-unnumbered** VSAs.

Example 2-15 shows how to redefine the VRF named *newyork* using the **ip:vrf-id** VSA.

Example 2-15 Redefining VRF Configurations

```
Change:
Cisco:Cisco-Avpair = "lcp:interface-config=ip vrf forwarding newyork"

To:
Cisco:Cisco-Avpair = "ip:vrf-id=newyork"
```

Example 2-16 shows how to redefine the Loopback 0 interface using the **ip:ip-unnumbered** VSA.

Example 2-16 Redefining IP Unnumbered Interfaces

```
Change:
Cisco:Cisco-Avpair = "lcp:interface-config=ip unnumbered Loopback 0"

To:
Cisco:Cisco-Avpair = "ip:ip-unnumbered=Loopback 0"
```

Placing PPPoA Sessions in Listening Mode

For better scalability and faster convergence of PPPoA, PPPoEoA, or LAC sessions, set sessions to passive mode, using the **atm pppatm passive** command in ATM subinterface configuration mode. This command places PPP or L2TP sessions on an ATM subinterface into listening mode. For large-scale PPP terminated aggregation (PPPoA and PPPoEoA) and L2TP (LAC), the **atm pppatm passive** command is required.

Instead of sending out Link Control Protocol (LCP) packets to establish the sessions actively, the sessions listen to the incoming LCP packets and become active only after they receive their first LCP packet. When PPPoX is in passive mode, the LAC brings up the sessions only when the subscribers become active and does not waste processing power polling all the sessions.

The following example configures passive mode for the PPPoA sessions on an ATM multipoint subinterface:

```
Router(config)# interface atm 1/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# range range-pppoa-1 pvc 100 199
Router(config-subif-atm-range)# encapsulation aal5mux ppp virtual-template 1
```

Scaling L2TP Tunnel Configurations

To prevent head-of-the-line blocking of the IP input process and save system resources, configure the following command in global configuration mode:

```
Router(config)# vpdn ip udp ignore checksum
```

When you configure this command, the router directly queues L2TP Hello packets and Hello acknowledgements to the L2TP control process. We recommend that you configure this command in all scaled LAC and LNS L2TP tunnel configurations.

If you do not configure the **vpdn ip udp ignore checksum** command, the L2TP software sends the packet to UDP to validate the checksum. When too many packets are queued to the IP input process, the router starts selective packet discard (SPD), which causes IP packets to be dropped.

Note Head-of-the-line blocking of the IP input process might occur in other non-L2TP configurations. A flush occurring on an input interface indicates that SPD is discarding packets.

Using the RADIUS Attribute cisco-avpair="lcp:interface-config"

When you use the **lcp:interface-config** RADIUS attribute to reconfigure the virtual-access subscriber interface, scaling on the Cisco 10000 series router decreases for the following reasons:

- The **lcp:interface-config** command syntax includes an IOS interface configuration command. This command is any valid IOS command that can be applied to an interface. When the **lcp:interface-config** attribute is downloaded from the RADIUS server to the Cisco 10000 series router, the command parser is activated to configure the interface as per AV-pair, determining if the option is valid and then applying the configuration to the virtual access interface (VAI).
- The **lcp:interface-config** command forces the Cisco 10000 series router to create full VAIs instead of subinterface VAIs. Full VAIs consume more memory and are less scalable, and they follow a significantly slower and different path when sessions are established.
- The **lcp:interface-config** command degrades the call rate.

To enhance the scalability of per-user configurations, in many cases different Cisco AV-pairs are available to place the subscriber interface in a virtual routing and forwarding (VRF) instance or to apply a policy map to the session. For example, use the **ip:vrf-id** and **ip:ip-unnumbered** VSAs to reconfigure the user's VRF. For more information, see the "Enhancing Scalability of Per-User Configurations" section on page 2-17.

Using Full Virtual Access Interfaces

A virtual access interface (VAI) is an interface that is dynamically created to terminate PPP subscribers. The Cisco router indicates full VAIs using a notation similar to **Virtual-Access6** (without a .number suffix).

Note For Cisco IOS Release 12.3(7)XI and later releases, the router does not support the use of full VAIs for broadband interfaces due to the scaling implications full VAIs have.

In general, the router creates full VAIs for one or more of the following reasons:

- Virtual template interface-specific configuration

  Some Cisco IOS configuration commands configured under the virtual template, such as the **carrier-delay** command, can force the router to create a full VAI. You can use the test command to determine the interface-specific configuration under the virtual template that triggered the full VAI.
- RADIUS attribute **lcp:interface-config**
- Global configuration **no virtual-template subinterface** command

Preventing Full Virtual Access Interfaces

The **lcp:interface-config** RADIUS attribute is used to reconfigure the subscriber interface. To accommodate the requirements of this attribute, the per-user authorization process forces the router to create full VAIs.

Cisco IOS Release 12.2(31)SB2, Release 12.2(28)SB6, and later releases include an enhancement that allows you to use the **lcp:interface-config** attribute while preserving subvirtual access subinterfaces. You can achieve this behaviour in the following ways:

- Entering the following command in global configuration mode to preserve virtual access subinterfaces:

  ```
  Router(config)# aaa policy interface-config allow-subinterface
  ```

- Sending a Cisco attribute-value pair (AV-pair) in the user's profile on the RADIUS server:

  ```
  cisco-avpair="lcp:interface-config allow-subinterface=yes"
  ```

When you use the **aaa policy interface-config allow-subinterface** command, the router does not allow you to reconfigure the router using any commands that interact with the interface's hardware interface descriptor block (HWIDB), for example, the **compression** command.

When you use the **lcp:interface-config** attribute, sessions are not established if the sessions receive the attribute and the attribute reconfigures the HWIDB for the virtual access interface (VAI).

When the **allow-subinterface=yes** option is used in the Cisco AV-pair or the **aaa policy interface-config allow-subinterface** command is set, enter the following command to verify the condition for which a full VAI reconfiguration is required:

```
Router# debug sss feature-name interface-config {error | event}
```

In general, for interface reconfiguration, use the dedicated Cisco vendor specific attributes (VSAs). For example, use **Cisco-Policy-Up** or **Cisco-Policy-Down,** or **ip:vrf-id** instead of **lcp:interface-config**. Alternatively, when no dedicated Cisco AV-pair is present, use **lcp:interface-config** with the **allow-subinterface=yes** option, or the **aaa policy interface-config allow-subinterface** command to preserve VAI subinterfaces (for example, to enable multicast on the subscriber interface).

CHAPTER 3

Configuring Remote Access to MPLS VPN

TheCisco 10000 series router supports the IP virtual private network (VPN) feature for Multiprotocol Label Switching (MPLS). MPLS-based VPNs allow service providers to deploy a scalable and cost-effective VPN service that provides a stable and secure path through the network. An enterprise or Internet service provider (ISP) can connect to geographically dispersed sites through the service provider's network. Using the MPLS backbone, a set of sites are interconnected to create an MPLS VPN.

The remote access (RA) to MPLS VPN feature on the Cisco 10000 series router allows the service provider to offer a scalable end-to-end VPN service to remote users. The RA to MPLS VPN feature integrates the MPLS-enabled backbone with broadband access capabilities. By integrating access VPNs with MPLS VPNs, a service provider can:

- Enable remote users and offices to seamlessly access their corporate networks
- Offer equal access to a set of different ISPs or retail service providers
- Integrate their broadband access networks with the MPLS-enabled backbone
- Provide an end-to-end VPN service to enterprise customers with remote access users and offices
- Separate network access and connectivity functions from ISP functions

The RA to MPLS VPN feature is described in the following topics:

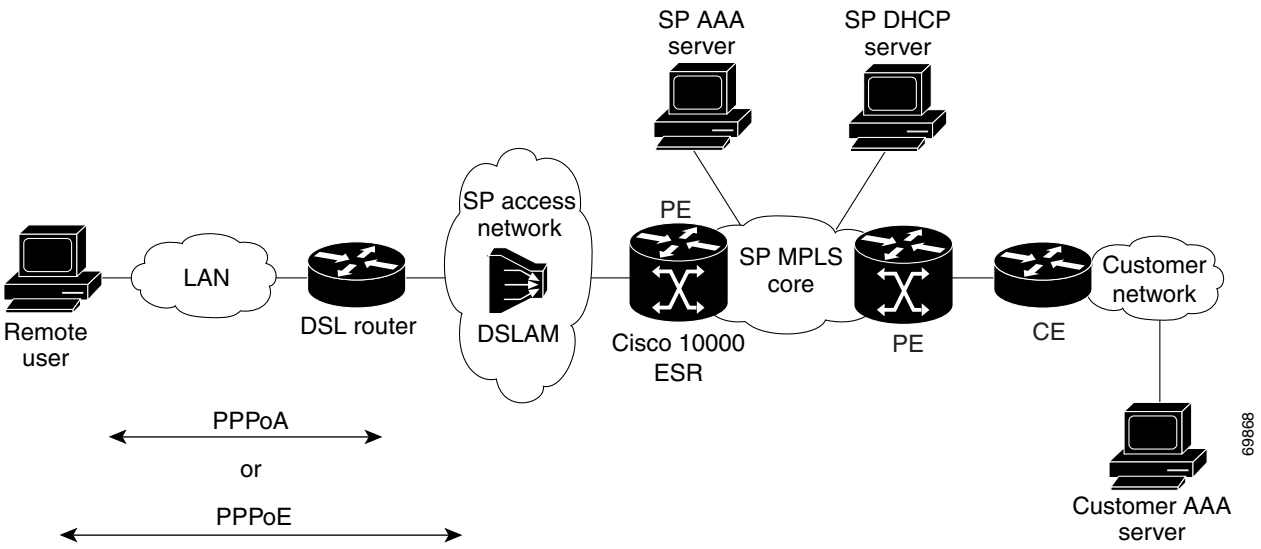
MPLS VPN Architecture

The MPLS VPN architecture enables the service provider to build the MPLS VPN network one time and add VPNs for new customers as needed, including them in the already established network. The elements that comprise the MPLS VPN are:

- Customer edge (CE) routers—The CPE devices to which subscribers in a customer's network connect. The CE router connects to a service provider's edge router (PE router). The CE router initiates the remote access session to the PE router.
- Provider edge (PE) routers—The router, such as the Cisco 10000 series router, located at the edge of the service provider's MPLS core network. The PE router connects to one or more CE routers and has full knowledge of the routes to the VPNs associated with those CE routers. The PE router does not have knowledge of the routes to VPNs whose associated CE routers are not connected to it.
- Provider (P) routers—The service provider routers that comprise the provider's core network. The P routers do not assign VPN information and they do not have any knowledge of CE routers. Instead, the main focus of the P router is on label switching.

Figure 3-1 shows an example of the MPLS VPN architecture.

Figure 3-1 MPLS VPN Network—Example



Access Technologies

The Cisco 10000 series router supports routed bridge encapsulation (RBE) protocol. Point-to-point protocol (PPP) access-based permanent virtual circuits (PVCs) is supported by using the following PPP access encapsulation methods:

- PPP over ATM (PPPoA)
- PPP over Ethernet (PPPoE)

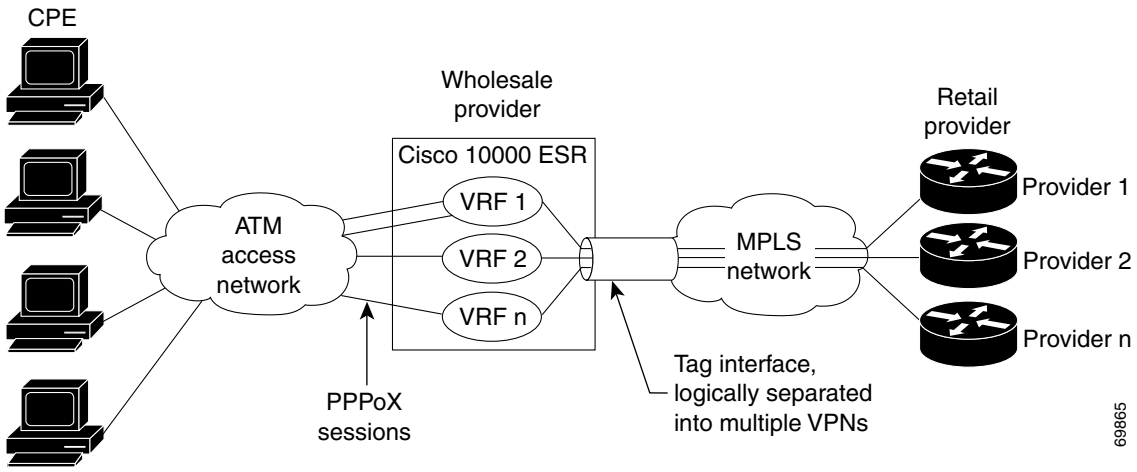
By using these PPP access technologies, the Cisco 10000 series router can terminate up to 32,000 sessions and support many features, including:

- Per session authentication based on Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP)
- Per session accounting
- Per session quality of service

Note The Cisco 10000 series router can terminate up to 32,000 ATM RBE sessions.

Figure 3-2 shows the topology of an integrated PPPoX (PPPoE or PPPoA) access to a multiprotocol label switching virtual private network (MPLS VPN) solution.

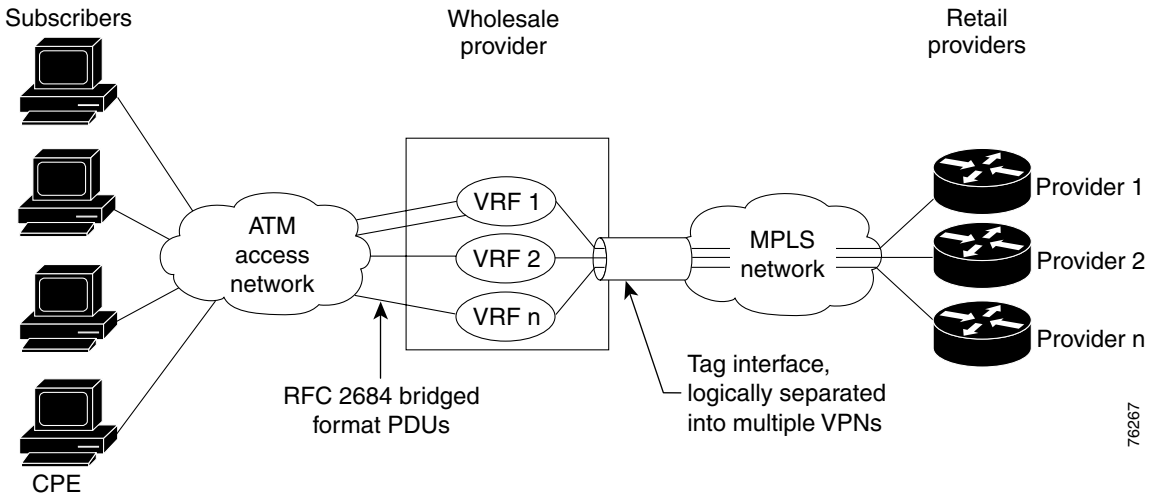
Figure 3-2 PPPoX Access to MPLS VPN Topology



In the figure, the service provider operates an MPLS VPN that interconnects all customer sites. The service provider's core network is an MPLS backbone with VPN service capability. The service provider provides all remote access operations to its customer. The network side interfaces are tagged interfaces, logically separated into multiple VPNs.

Figure 3-3 shows the topology of an RBE to MPLS VPN solution.

Figure 3-3 RBE to MPLS VPN Topology



In the figure, the wholesale provider uses VPNs to separate the subscribers of different retail providers. The subscribers are uniquely placed in VRFs on the access side. A tag interface separates traffic for the different retail providers on the network side. The MPLS VPN technology is used to assign tags in a VPN-aware manner.

PPP over ATM to MPLS VPN

The Cisco 10000 series router supports a PPP over ATM (PPPoA) connection to an MPLS VPN architecture. In this model, when a remote user attempts to establish a connection with a corporate network, a PPPoA session is initiated and is terminated on the service provider's virtual home gateway (VHG) or provider edge (PE) router. All remote hosts connected to a particular CE router must be part of the same VPN to which the CE router is connected.

The following events occur when the remote user attempts to access the corporate network or ISP:

1. A PPPoA session is initiated over the broadband access network.
2. The VHG/PE router accepts and terminates the PPPoA session.
3. The VHG/PE router obtains virtual access interface (VAI) configuration information.
   a. The VHG/PE obtains virtual template interface configuration information, which typically includes virtual routing and forwarding (VRF) mapping for sessions.
   b. The VHG/PE sends a separate request to either the customer's or service provider's RADIUS server for the VPN to authenticate the remote user.
   c. The VPN's VRF instance was previously instantiated on the VHG or PE. The VPN's VRF contains a routing table and other information associated with a specific VPN.

   Typically, the customer RADIUS server is located within the customer VPN. To ensure that transactions between the VHG/PE router and the customer RADIUS server occur over routes within the customer VPN, the VHG/PE router is assigned at least one IP address that is valid within the VPN.

4. The VHG/PE router forwards accounting records to the service provider's proxy RADIUS server, which in turn logs the accounting records and forwards them to the appropriate customer RADIUS server.
5. The VHG/PE obtains an IP address for the CPE. The address is allocated from one of the following:
   - Local address pool
   - Service provider's RADIUS server, which either specifies the address pool or directly provides the address
   - Service provider's DHCP server
6. The CPE is now connected to the customer VPN. Packets can flow to and from the remote user.

Use virtual template interfaces to map sessions to VRFs. The Cisco 10000 series router can then scale to 32,000 sessions. In Cisco IOS Release 12.2(16)BX1 and later releases, when you map sessions to VRFs by using the RADIUS server, use the syntax **ip:vrf-id** or **ip:ip-unnumbered**. These vendor specific attributes (VSAs) enhance the scalability of per-user configurations because a new full virtual access interface is not required. For more information, see the "Enhancing Scalability of Per-User Configurations" section on page 2-17.

Note In releases earlier than Cisco IOS Release 12.2(16)BX1, to map sessions to VRFs by using the RADIUS server, use the syntax **lcp:interface-config**. This configuration forces the Cisco 10000 series router to use full access virtual interfaces, which decreases scaling. We recommend that you do not use this configuration. Upgrading to Cisco IOS Release 12.2(16)BX1 or later eliminates this restriction.

PPP over Ethernet to MPLS VPN

The Cisco 10000 series router supports a PPP over Ethernet (PPPoE) connection to an MPLS VPN architecture. In this model, when a remote user attempts to establish a connection with a corporate network, a PPPoE session is initiated and is terminated on the service provider's virtual home gateway (VHG) or provider edge (PE) router. All remote hosts connected to a particular CE router must be part of the VPN to which the CE router is connected.

The PPPoE to MPLS VPN architecture is a flexible architecture with the following characteristics:

- A remote host can create multiple concurrent PPPoE sessions, each to a different VPN.
- If multiple remote hosts exist behind the same CE router, each remote host can log in to a different VPN.
- Any remote host can log in to any VPN at any time because each VHG or PE router has the VRFs for all possible VPNs pre-instantiated on it. This configuration requires that the VRF be applied through the RADIUS server, which can cause scalability issues (see the following note).

Use virtual template interfaces to map sessions to VRFs. The Cisco 10000 series router can then scale to 32,000 sessions. In Cisco IOS Release 12.2(16)BX1 and later releases, when you map sessions to VRFs by using the RADIUS server, use the syntax **ip:vrf-id** or **ip:ip-unnumbered**. These vendor specific attributes (VSAs) enhance the scalability of per-user configurations because a new full virtual access interface is not required. For more information, see the "Enhancing Scalability of Per-User Configurations" section on page 2-17.

Note For releases earlier than Cisco IOS Release 12.2(16)BX1, to map sessions to VRFs by using the RADIUS server, use the syntax **lcp:interface-config**. This configuration forces the Cisco 10000 series router to use full access virtual interfaces, which decreases scaling. We recommend that you do not use this configuration. Upgrading to Cisco IOS Release 12.2(16)BX1 or later releases will eliminate this restriction.

The following events occur as the VHG or PE router processes the incoming PPPoE session:

1. A PPPoE session is initiated over the broadband access network.
2. The VHG/PE router accepts and terminates the PPPoE session.
3. The VHG/PE router obtains virtual access interface (VAI) configuration information.
   a. The VHG/PE obtains virtual template interface configuration information, which typically includes VRF mapping for sessions.
   b. The VHG/PE sends a separate request to either the customer's or service provider's RADIUS server for the VPN to authenticate the remote user.
   c. The VPN's VRF instance was previously instantiated on the VHG or PE. The VPN's VRF contains a routing table and other information associated with a specific VPN.

   Use virtual template interfaces to map sessions to VRFs. The Cisco 10000 series router can then scale to 32,000 sessions. In Cisco IOS Release 12.2(16)BX1 and later releases, when you map sessions to VRFs by using the RADIUS server, use the syntax **ip:vrf-id** or **ip:ip-unnumbered**. These vendor specific attributes (VSAs) enhance the scalability of per-user configurations because a new full virtual access interface is not required. For more information, see the "Enhancing Scalability of Per-User Configurations" section on page 2-17.

   Note For releases earlier than Cisco IOS Release 12.2(16)BX1, to map sessions to VRFs by using the RADIUS server, use the syntax **lcp:interface-config**. This configuration forces the Cisco 10000 series router to use full access virtual interfaces, which decreases scaling. We recommend that you do not use this configuration. Upgrading to Cisco IOS Release 12.2(16)BX1 or later releases will eliminate this restriction.

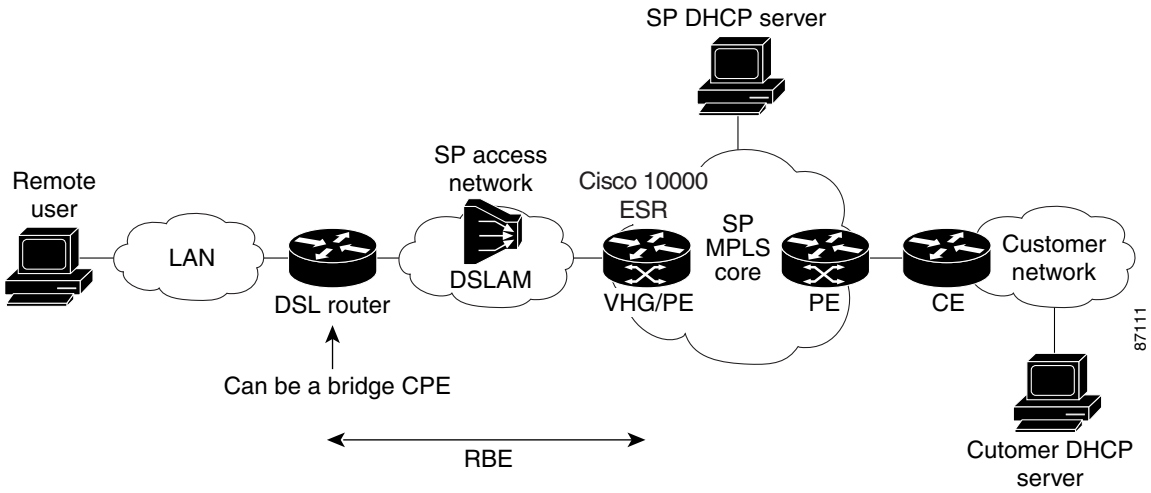
   Typically, the customer RADIUS server is located within the customer VPN. To ensure that transactions between the VHG/PE router and the customer RADIUS server occur over routes within the customer VPN, the VHG/PE router is assigned at least one IP address that is valid within the VPN.
4. The VHG/PE router forwards accounting records to the service provider's proxy RADIUS server, which in turn logs the accounting records and forwards them to the appropriate customer RADIUS server.
5. The VHG/PE obtains an IP address for the CPE. The address is allocated from one of the following:
   - Local address pool
   - Service provider's RADIUS server, which either specifies the address pool or directly provides the address
   - Service provider's DHCP server
6. The CPE is now connected to the customer VPN. Packets can flow to and from the remote user.

RBE over ATM to MPLS VPN

The Cisco 10000 series router supports an ATM RBE to MPLS VPN connection. RBE is used to route IP over bridged RFC 1483 Ethernet traffic from a stub-bridged LAN. The ATM connection appears like a routed connection; however, the packets received on the interface are bridged IP packets. RBE looks at the IP header of the packets arriving at an ATM interface and routes the packets instead of bridging them.

In Figure 3-4, RBE is configured between the DSL router and the Cisco 10000 series router, acting as the VHG/PE router.

Figure 3-4 DSL RBE to MPLS VPN Integration



The DSL router can be set up as a pure bridge or it can be set up for integrated routing and bridging (IRB) where multiple LAN interfaces are bridged through the bridge group virtual interface (BVI). Each of the DSL routers terminates on a separate point-to-point subinterface on the VHG/PE, which is statically configured with a specific VRF. Remote user authentication or authorization is available with Option 82 for DSL RBE remote access. RBE treats the VHG/PE subinterface as if it is connected to an Ethernet LAN, but avoids the disadvantages of pure bridging, such as broadcast storms, IP hijacking, and ARP spoofing issues. Address management options include static and VRF-aware DHCP servers.

Note For more information, see the "DSL Access to MPLS VPN Integration" chapter in the *Cisco Remote Access to MPLS VPN Solution Overview and Provisioning Guide, Release 2.0*, located at the following URL.
http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/rampls2/ovprov/ra_op_05.htm

MPLS VPN ID

The MPLS VPN ID is a 14-digit hexadecimal number that uniquely identifies a VPN and its associated VRF across all VHGs and PE routers in the network. In a router with multiple VPNs configured, you can use a VPN ID to identify a particular VPN. The VPN ID follows a standard specification (RFC 2685). The configuration of a VPN ID is optional.

You can configure a VRF instance for each VPN configured on the Cisco 10000 series router. By using the **vpn id** VRF configuration command, you can assign a VPN ID to a VPN. The router stores the VPN ID in the corresponding VRF structure for the VPN (see the "Configuring Virtual Routing and Forwarding Instances" section on page 3-13).

Note The VPN ID is used for provisioning only. BGP routing updates do not include the VPN ID.

DHCP servers use the VPN ID to identify a VPN and allocate resources as the following describes:

1. A VPN DHCP client requests a connection to the Cisco 10000 series router (PE router) from a VRF interface.
2. The PE router determines the VPN ID associated with that interface.
3. The PE router sends a request with the VPN ID and other information for assigning an IP address to the DHCP server.
4. The DHCP server uses the VPN ID and IP address information to process the request.
5. The DHCP server sends a response back to the PE router, allowing the VPN DHCP client access to the VPN.

The RADIUS server uses the VPN ID to assign dialin users to the proper VPN. Typically, a user login consists of the following packets:

- Access-Request packet—A query from the network access server (NAS) that contains the user name, encrypted password, NAS IP address, VPN ID, and port. The format of the request also provides information on the type of session that the user wants to initiate.
- Access-Accept or Access-Reject packet—A response from the RADIUS server. The server returns an Access-Accept response if it finds the user name and verifies the password. The response includes a list of attribute-value (AV) pairs that describe the parameters to be used for this session. If the user is not authenticated, the RADIUS server returns an Access-Reject packet, and access is denied.

Note For more information, see the *MPLS VPN ID, Release 12.2(4)B* feature module, located at the following URL.
http://www.cisco.com/en/US/docs/ios/12_2/12_2b/12_2b4/feature/guide/12b_vpn.html

DHCP Relay Agent Information Option—Option 82

The Cisco 10000 series router supports the Dynamic Host Configuration Protocol (DHCP) relay agent information option (Option 82) feature when ATM routed bridge encapsulation (RBE) is used to configure DSL access. This feature communicates information to the DHCP server by using a suboption of the DHCP relay agent information option called *agent remote ID*. The information sent in the agent remote ID includes an IP address identifying the relay agent, information about the ATM interface, and information about the PVC over which the DHCP request came in. The DHCP server can use this information to make IP address assignments and security policy decisions.

Acting as the DHCP relay agent, the Cisco 10000 series router can also include VPN ID information in the agent remote ID suboption when forwarding client-originated DHCP packets to a DHCP server that has knowledge of existing VPNs. The VPN-aware DHCP server receives the DHCP packets and uses the VPN ID information to determine from which VPN to allocate an address. The DHCP server responds to the DHCP relay agent and includes information that identifies the originating client.

Note For more information, see the *DHCP Option 82 Support for Routed Bridge Encapsulation, Release 12.2(2)T* feature module.

DHCP Relay Support for MPLS VPN Suboptions

The DHCP relay agent information option (Option 82) enables a Dynamic Host Configuration Protocol (DHCP) relay agent to include information about itself when forwarding client-originated DHCP packets to a DHCP server. In some environments, the relay agent has access to one or more MPLS VPNs. A DHCP server that wants to offer service to DHCP clients on those different VPNs needs to know the VPN where each client resides. The relay agent typically knows about the VPN association of the DHCP client and includes this information in the relay agent information option.

The DHCP relay support for MPLS VPN suboptions feature allows the Cisco 10000 series router, acting as the DHCP relay agent, to forward VPN-related information to the DHCP server by using the following three suboptions of the DHCP relay agent information option:

- VPN identifier
- Subnet selection
- Server identifier override

The DHCP relay agent uses the VPN identifier suboption to tell the DHCP server the VPN for each DHCP request that it passes on to the DHCP server, and also uses the suboption to properly forward any DHCP reply that the DHCP server sends back to the relay agent. The VPN identifier suboption contains the VPN ID configured on the incoming interface to which the client is connected. If you configure the VRF name but not the VPN ID, the VRF name is used as the VPN identifier suboption. If the interface is in global routing space, the router does not add the VPN suboptions.

The subnet selection suboption allows the separation of the subnet where the client resides from the IP address that is used to communicate with the relay agent. In some situations, the relay agent needs to specify the subnet on which a DHCP client resides that is different from the IP address the DHCP server can use to communicate with the relay agent. The DHCP relay agent includes the subnet selection suboption in the relay agent information option, which the relay agent passes on to the DHCP server.

The server identifier override suboption contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. By using this information, the DHCP client sends all renew and release packets to the relay agent. The relay agent adds all the VPN suboptions and then forwards the renew and release packets to the original DHCP server.

After adding these suboptions to the DHCP relay agent information option, the gateway address changes to the relay agent's outgoing interface on the DHCP server side. The DHCP server uses this gateway address to send reply packets back to the relay agent. The relay agent then removes the relay agent information options and forwards the packets to the DHCP client on the correct VPN.

Note For more information, see the *DHCP Relay Support for MPLS VPN Suboptions, Release 12.2(4)B* feature module, located at the following URL.
http://www.cisco.com/en/US/docs/ios/12_2/12_2b/12_2b4/feature/guide/12b_dhc.html

Feature History for RA to MPLS VPN

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was integrated into Cisco IOS Release 12.2(4)BZ1.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for RA to MPLS VPN

The RA to MPLS VPN feature has the following restrictions:

- When BGP aggregates customer routes, the received packets that match the aggregate route require an additional feedback in the PXF forwarding engine, which reduces performance.
- RBE to MPLS VPN does not support MAC-layer access lists; only IP access lists are supported.
- Before configuring DHCP relay support for MPLS VPN suboptions, you must configure standard MPLS VPNs. For more information, see the "Configuring Virtual Private Networks" section on page 3-28 and the "Configuring the MPLS Core Network" section on page 3-12, or see the *Cisco IOS Switching Services Configuration Guide, Release 12.2*, located at the following URL http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/fswtch_c.html.
- The VPN ID is not used to control the distribution of routing information or to associate IP addresses with VPN IDs in routing updates.

Prerequisites for RA to MPLS VPN

The RA to MPLS VPN feature has the following requirements:

- Your network must be running the following Cisco IOS services before you configure VPN operation:
  - MPLS in the service provider backbone routers
  - Tag distribution protocol (TDP) or the label distribution protocol (LDP)
  - BGP in all routers providing a VPN service
  - Cisco Express Forwarding (CEF) switching in each MPLS-enabled router

Note IP CEF is on by default on the Cisco 10000 series router and it cannot be turned off. If you attempt to enable IP CEF, an error appears.

- For PPPoX to MPLS VPN networks, the Cisco 10000 series router must be running Cisco IOS Release 12.2(4)BZ1 or later releases and the performance routing engine must be installed in the router's chassis.
- For ATM RBE to MPLS VPN networks, the Cisco 10000 series router must be running Cisco IOS Release 12.2(15)BX or later releases and the performance routing engine must be installed in the router's chassis.
- You must configure DHCP option 82 support on the DHCP relay agent by using the **ip dhcp relay information option** command before you can use the DHCP Option 82 support for the RBE feature.
- Configure all the PE routers that belong to the same VPN with the same VPN ID. Make sure that the VPN ID is unique to the service provider network.

Configuration Tasks for RA to MPLS VPN

To configure the RA to MPLS VPN feature, perform the following configuration tasks:

Configuring the MPLS Core Network

To configure an MPLS core network, perform the following tasks:

Enabling Label Switching of IP Packets on Interfaces

Enable label switching of IP packets on each PE router interface on the MPLS side of the network. The Cisco 10000 series router MPLS network side interface is a tagged interface. The packets passing through the interface are tagged packets.

Note Multiple interfaces require a Label Switch Router (LSR).

To enable label switching of IP packets on interfaces, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# mpls ip`	Enables label switching of IP packets on the interface.

Note The Cisco 10000 series router supports the PPP Terminated Aggregation (PTA) to VRF feature, which terminates incoming PPP sessions and places them into the appropriate VRF for transport to the customer network. Unlike the RA to MPLS VPN model, the network side interface is not a tagged interface and there are no tagged packets. In the PTA to VRF model, the network side interface is an IP interface with IP packets. In this case, the traffic for the different VRFs is typically separated at Layer 2.

Configuring Virtual Routing and Forwarding Instances

Configure VRF instances on each PE router in the provider network. Create one VRF for each VPN connected using the **ip vrf** command in global configuration mode or router configuration mode.

To create the VRF, do the following:

- Specify the correct route distinguisher (RD) used for that VPN using the **rd** command in VRF configuration submode. The RD is used to extend the IP address so that you can identify the VPN to which it belongs.
- Set up the import and export policies for the MP-BGP extended communities using the **route-target** command in VRF configuration submode. These policies are used for filtering the import and export process.

To configure a VRF, enter the following commands on the PE router beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the virtual routing instance by assigning a VRF name.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.
Step 3	`Router(config-vrf)# route-target {import | export | both} route-target-ext-community`	Creates a list of import and export route target communities for the specified VRF.
Step 4	`Router(config-vrf)# vpn id oui:vpn-index`	Assigns or updates a VPN ID on the VRF. The VPN ID uniquely identifies a VPN and VRF across all VHG and PE routers in the network. **Note** The VPN ID is used for provisioning only. BGP routing updates do not include the VPN ID.

Associating VRFs

After you define and configure the VRFs on the PE routers, associate each VRF with:

- An interface or subinterface
- A virtual template interface

The virtual template interface is used to create and configure a virtual access interface (VAI). For information about configuring a virtual template interface, see the "Configuring a Virtual Template Interface" section on page 3-17.

To associate a VRF, enter the following commands on the PE router beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with an interface or subinterface.
Step 2	`Router(config-if)# ip address ip-address mask`	Sets a primary or secondary address for an interface.
Step 3	`Router(config-if)# exit`	Returns to global configuration mode.

	Command	Purpose
Step 4	`Router(config)# interface virtual-template number`	Creates a virtual template interface and enters interface configuration mode.
Step 5	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with a virtual template interface.

Note Apply the **ip vrf forwarding** command and then the **ip address** command. If you do not, the **ip vrf forwarding** command removes the existing IP address on the interface.

Example 3-1 Associating a VRF with an Interface

```
interface GigabitEthernet7/0/0.1
 encapsulation dot1Q 11
 ip vrf forwarding vpn1
 ip address 192.168.1.1 255.255.255.0
!
```

Example 3-2 Associating a VRF with a Virtual Template Interface

```
interface Virtual-Template1
 ip vrf forwarding vpn1
 ip unnumbered Loopback1
 no peer default ip address
 ppp authentication chap vpn1
 ppp authorization vpn1
 ppp accounting vpn1
```

Configuring Multiprotocol BGP PE to PE Routing Sessions

To configure multiprotocol BGP (MP-BGP) routing sessions between the PE routers, enter the following commands on the PE routers beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system`	Configures the internal BGP (iBGP) routing process with the autonomous system number passed along to other iBGP routers.
Step 2	`Router(config-router)# no bgp default ipv4-unicast`	Disables IPv4 BGP routing.
Step 3	`Router(config-router)# neighbor {ip-address | peer-group-name} remote-as as-number`	Configures the neighboring PE router's IP address or iBGP peer group and identifies it to the local autonomous system. The MP-BGP neighbors must use the loopback addresses.
Step 4	`Router(config-router)# neighbor {ip-address | peer-group-name} update-source interface-type`	Allows iBGP sessions to use any operational interface for TCP connections.
Step 5	`Router(config-router)# neighbor {ip-address | peer-group-name} activate`	Activates route exchanges with the global BGP neighbors.

	Command	Purpose
Step 6	`Router(config-router)# address-family ipv4 vrf vrf-name`	Enters address family configuration mode and configures the VRF routing table for BGP routing sessions that use standard IPv4 address prefixes. The *vrf-name* argument specifies the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	`Router(config-router-af)# redistribute protocol`	Redistributes routes from one routing domain into another routing domain. The *protocol* argument is the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **ospf**, **static** [**ip**], or **rip**. The **connected** keyword refers to routes that are established automatically by virtue of having enabled IP on an interface.
Step 8	`Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	`Router(config-router)# address-family vpnv4 [unicast]`	Enters address family configuration mode for configuring BGP routing sessions that use standard Virtual Private Network (VPN) Version 4 address prefixes. (Optional) The **unicast** keyword specifies VPN Version 4 unicast address prefixes.
Step 10	`Router(config-router-af)# neighbor {ip-address | peer-group-name} activate`	Activates route exchanges with the global BGP neighbors.
Step 11	`Router(config-router-af)# neighbor {ip-address | peer-group-name} send-community [both]`	Specifies that a communities attribute should be sent to a BGP neighbor. The **both** keyword specifies that both communities attributes should be sent.

Example 3-3 Configuring MP-BGP

```
router bgp 100
    no synchronization
    no bgp default ipv4-unicast
    bgp log-neighbor-changes
    neighbor 10.1.1.4 remote-as 100
    neighbor 10.1.1.4 update-source Loopback0
    neighbor 10.1.1.4 activate
    neighbor 10.3.1.4 remote-as 100
    neighbor 10.3.1.4 update-source Loopback0
    neighbor 10.3.1.4 activate
    no auto-summary
!
address-family ipv4 vrf vrf-1
    redistribute connected
    no auto-summary
    no synchronization
    exit-address-family
!
```

```
address-family vpnv4
    neighbor 10.1.1.4 activate
    neighbor 10.1.1.4 send-community both
    neighbor 10.3.1.4 activate
    neighbor 10.3.1.4 send-community both
    exit-address-family
!
```

Note Typically, you enable BGP only on the PE routers. It is not necessary to enable BGP on all provider (P) core routers. However, if your network topology includes a route reflector, you may then enable BGP on a core router, which might be a P or PE router.

Configuring Access Protocols and Connections

The Cisco 10000 series router supports the following access protocols:

- PPP over ATM
- PPP over Ethernet
- RBE over ATM

When a remote user initiates a PPPoA or PPPoE session to the Cisco 10000 series router, a predefined configuration template is used to configure a virtual interface known as a virtual access interface (VAI). The VAI is created and configured dynamically by using a virtual template interface. When the user terminates the session, the VAI goes down and the resources are freed for other client uses.

Note Virtual template interfaces and VAIs do not apply to RBE over ATM.

The virtual template interface is a logical entity that the Cisco 10000 series router applies dynamically as needed to a connection. It is a configuration for an interface, but it is not tied to the physical interface. The VAI uses the attributes of the virtual template to create the session, which results in a VAI that is uniquely configured for a specific user.

After you configure a virtual template, configure the virtual connection that will use the template and then apply the template to the connection. The order in which you create virtual templates and configure the virtual connections that use the templates is not important. However, both the virtual templates and connections must exist before a remote user initiates a session to the Cisco 10000 series router.

The following sections describe how to create a virtual template and apply it to a VAI. For more information, see the "Configuring Virtual Template Interfaces" chapter in the *Cisco IOS Dial Technologies Configuration Guide, Release 12.2*.

Note If you are using a RADIUS server, the RADIUS configuration takes precedence over the virtual template interface configuration. For example, the RADIUS configuration might override a number of parameters with the remainder of the configuration coming from the virtual template interface.

To configure access protocols and connections, perform the following configuration tasks. The first task listed is required and you can perform any of the remaining tasks as needed:

- Configuring a Virtual Template Interface, page 3-17
- Configuring PPP over ATM Virtual Connections and Applying Virtual Templates, page 3-18
- Configuring PPPoE over ATM Virtual Connections and Applying Virtual Templates, page 3-18

- Configuring PPPoE over Ethernet Virtual Connections and Applying Virtual Templates, page 3-20
- Configuring RBE over ATM Virtual Connections, page 3-22

Configuring a Virtual Template Interface

To create and configure a virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config)# ip unnumbered ethernet` *number*	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# ppp authentication chap`	Enables PPP authentication on the virtual template interface.
Step 4	`Router(config-if)# ppp ipcp ip address required`	Required for legacy dial up and DSL networks. Prevents a PPP session from being set up with 0.0.0.0 remote ip address.

Example 3-4 Configuring a Virtual Template Interface

```
interface virtual-template 1
ip unnumbered Loopback1
no peer default ip address
ppp authentication chap vpn1
ppp ipcp ip address required
ppp authorization vpn1
ppp accounting vpn1
```

Monitoring and Maintaining a Virtual Access Interface

When a virtual template interface is applied dynamically to an incoming user session, a virtual access interface (VAI) is created. You cannot use the command line interface (CLI) to directly create or configure a VAI, but you can display and clear the VAI by using the following commands in privileged EXEC mode:

Command	Purpose
`Router# show interfaces virtual-access` *number* [**configuration**]	Displays the configuration of the active VAI that was created using a virtual template interface. The **configuration** keyword restricts output to configuration information.
`Router# clear interface virtual-access` *number*	Tears down the live sessions and frees the memory for other client uses.

Example 3-5 Displaying the Active VAI Configuration

```
Router# show interfaces virtual-access 1.1 configuration
!
interface virtual-access1.1
    ip vrf forwarding vrf-1
    ip unnumbered Loopback1
    no ip proxy-arp
```

```
peer default ip address pool vrf-1
ppp authentication chap
end
```

Note Virtual-access 1.1 is a PPPoE subinterface.

Example 3-6 Clearing Live Sessions

```
Router# clear interface virtual-access 1.1
Router#
```

Configuring PPP over ATM Virtual Connections and Applying Virtual Templates

To configure a range of PVC connections and apply a virtual template interface to them, perform the following configuration task:

- Configuring Encapsulated PPP over ATM Permanent Virtual Circuits, page 3-18

Note For more information, see the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide, Release 12.2*.

Configuring Encapsulated PPP over ATM Permanent Virtual Circuits

Configure ATM permanent virtual circuits (PVCs) for encapsulated PPP over ATM on either point-to-point or multipoint subinterfaces. Using point-to-multipoint PVCs significantly increases the maximum number of PPPoA sessions that you can run on the Cisco 10000 series router.

To configure a PVC range with encapsulated PPPoA, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm` *slot/port.subinterface-number* `multipoint`	Specifies an ATM multipoint subinterface.
Step 2	`Router(config-subif)# range` [*range-name*] `pvc` *start-vpi/start-vci end-vpi/end-vci*	Creates a range of PVCs.
Step 3	`Router(config-if-atm-range)# encapsulation aal5`*encap* `ppp virtual-template` *number*	Configures the ATM adaptation layer (AAL) and encapsulation type on an ATM PVC range and links it to the virtual template interface.

Configuring PPPoE over ATM Virtual Connections and Applying Virtual Templates

To configure PPPoE over ATM, perform the following configuration tasks:

- Configure a virtual template (see the "Configuring a Virtual Template Interface" section on page 3-17).
- Configuring a VPDN Group for PPPoE over ATM, page 3-19
- Configuring PPPoE on ATM Permanent Virtual Circuits, page 3-19
- Configuring PPPoE on ATM PVCs Using a Different MAC Address, page 3-20

Note For more information, see the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide, Release 12.2*.

Configuring a VPDN Group for PPPoE over ATM

To configure the physical interface that will carry the PPPoE session and link it to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables virtual private dial network (VPDN) configuration on this router.
Step 2	`Router(config)# vpdn group` *name*	Associates a VPDN group with a customer or VPDN profile.
Step 3	`Router(config-vpdn)# accept-dialin`	Creates an accept dial-in VPDN group.
Step 4	`Router(config-vpdn-acc-in)# protocol pppoe`	Specifies the VPDN group to be used to establish PPPoE sessions.
Step 5	`Router(config-vpdn-acc-in)# virtual-template` *template-number*	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 6	`Router(config-vpdn)# pppoe limit per-vc` *number*	Specifies the maximum number of PPPoE sessions to be established over a virtual circuit.

Configuring PPPoE on ATM Permanent Virtual Circuits

To configure PPPoE on a range of ATM PVCs, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm` *slot/0.subinterface-number* `multipoint`	Specifies an ATM multipoint subinterface.
Step 2	`Router(config-subif)# range` [*range-name*] `pvc` *start-vpi/start-vci end-vpi/end-vci*	Creates a range of PVCs.
Step 3	`Router(config-if-atm-range)# encapsulation aal5snap`	Configures VC multiplexed encapsulation on a PVC range.
Step 4	`Router(config-if)# protocol pppoe`	Specifies the VPDN group to be used to establish PPPoE sessions on the PVC range.

Configuring PPPoE on ATM PVCs Using a Different MAC Address

To change the way PPPoE selects a MAC address when PPPoE and RBE are configured on two separate PVCs on the same DSL line, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group pppoe-term`	Specifies the VPDN group to be used to establish PPPoE sessions on a PVC.
Step 2	`Router(config-vpdn)# accept-dialin`	Configures the L2TP access concentrator (LAC) to accept PPPoE sessions from a client and creates an accept-dialin VPDN subgroup.
Step 3	`Router(config-vpdn-acc-in)# protocol pppoe`	Configures a static map for an ATM PVC.
Step 4	`Router(config-vpdn-acc-in)# exit`	Exits accept-dialin configuration mode and returns to VPDN configuration mode.
Step 5	`Router(config-vpdn)# pppoe mac-address {autoselect | mac-address}`	Changes the way PPPoE selects a MAC address. The autoselect option always chooses a "MAC plus 7" address and no other address. For example, it chooses the ATM interface MAC address, interface MAC address plus 1, plus 2, plus 3, plus 4, plus 5, or plus 6). Use the *mac-address* option to enter an explicit MAC address value.

Note Use the **pppoe mac-address** command in VPDN group configuration mode. The Cisco 10000 series router applies the command to all PPPoEoA sessions brought up after you issue the command. MAC address usage does not change until you explicitly configure it using the **pppoe mac-address** command. The router limits the change to PPPoE sessions on ATM interfaces only and does not apply it to other interfaces on which PPPoE operates (such as Ethernet, Ethernet VLAN and DOCSIS interfaces).

Configuring PPPoE over Ethernet Virtual Connections and Applying Virtual Templates

To configure PPPoE over Ethernet, perform the following configuration tasks:

- Configuring a Virtual Template Interface, page 3-17
- Configuring PPPoE over Ethernet in a BBA Group, page 3-21

Configuring PPPoE over Ethernet in a BBA Group

Note Cisco IOS Release 12.2(15)BX does not support RADIUS configuration of BBA groups. You must configure BBA groups manually.

To configure a broadband aggregation (BBA) group for PPPoE and to link it to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# bba-group pppoe {name | global}`	Configures a BBA group to be used to establish PPPoE sessions. *name* identifies the BBA group. You can have multiple BBA groups. **global** is the default BBA group used for ATM connections when a BBA group name is not specified.
Step 2	`Router(config-bba)# virtual-template template-number`	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 3	`Router(config-bba)# pppoe limit per-mac per-mac-limit`	(Optional) Specifies the maximum number of sessions per MAC address for each PPPoE port that uses the group.
Step 4	`Router(config-bba)# pppoe limit max-sessions number`	(Optional) Specifies the maximum number of PPPoE sessions that can be terminated on this router from all interfaces.
Step 5	`Router(config-bba)# pppoe limit per-vc per-vc-limit`	(Optional) Specifies the maximum number of PPPoE sessions for each VC that uses the group.
Step 6	`Router(config-bba)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# interface atm slot/subslot/port.subinterface`	Specifies the interface to which you want to attach the BBA group.
Step 8	`Router(config-if)# pvc [name] vpi/vci`	Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration mode. (Optional) *name* specifies the name of the PVC or map. The name can be up to 16 characters. *vpi/* specifies the ATM network VPI for the PVC that you named. Valid values are from 0 to 255. If a value is not specified, the vpi value is set to 0. *vci* specifies the ATM network VCI for the PVC you named. Valid values are from 0 to 1 less than the maximum value set for this interface using the **atm vc-per-vp** command. **Note** You cannot set both *vpi* and *vci* to 0; if one is 0, the other cannot be 0.
Step 9	`Router(config-if)# protocol pppoe group group-name`	Attaches the BBA group to the PVC.

Note You cannot simultaneously configure a BBA group for PPPoE and a VPDN group for PPPoE. If you configure a BBA group and then you configure a VPDN group, the **protocol** command in VPDN accept-dialin configuration mode does not include an option for PPPoE (for example, you cannot specify the **protocol pppoe** command). Use the **no bba-group pppoe** command to re-enable the **pppoe** option for the **protocol** command.

Configuring RBE over ATM Virtual Connections

To configure RBE over ATM virtual connections and apply virtual templates, perform the following configuration tasks:

- Configuring the PE Router, page 3-22
- Configuring DHCP Option 82 for RBE, page 3-25
- Configuring DHCP Relay Support for MPLS VPN Suboptions, page 3-26
- Specifying a VPN ID, page 3-27

Configuring the PE Router

To configure the PE router, perform the following required configuration tasks:

- Defining Loopbacks, page 3-22
- Defining PVCs, page 3-23
- Configuring Label Switching, page 3-23
- Configuring the VRF for Each VPN, page 3-23
- Configuring a Dedicated PVC, page 3-24
- Configuring BGP to Advertise Networks, page 3-24

Note For more information, see the "DSL Access to MPLS VPN Integration" chapter in the *Cisco Remote Access to MPLS VPN Solution Overview and Provisioning Guide, Release 2.0*.

Defining Loopbacks

To define loopbacks, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback number`	Creates a loopback interface to reach the router. Enters interface configuration mode.
Step 2	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF with the loopback interface.
Step 3	`Router(config-if)# ip address [address] [netmask]`	Assigns an IP address to the loopback interface.

Defining PVCs

To define PVCs, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port.subinterface-number point-to-point`	Specifies an ATM point-to-point subinterface. Enters subinterface configuration mode.
Step 2	`Router(config-subif)# ip vrf forwarding vrf-name`	Associates a VRF with the ATM point-to-point subinterface.
Step 3	`Router(config-subif)# ip unnumbered Loopback number`	Configures the ATM subinterface as unnumbered to a loopback interface. **Note** The loopback interface must be in the same VRF.
Step 4	`Router(config-subif)# pvc [vpi/vci number]`	Configures the PVC on the subinterface. Enters PVC configuration mode.
Step 5	`Router(config-subif-pvc)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type on the ATM PVC.
Step 6	`Router(config-subif-pvc)# no protocol ip inarp`	Disables Inverse ARP on the ATM PVC.

Configuring Label Switching

To configure label switching on the interface connected to the MPLS cloud, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port.subinterface-number tag-switching`	Connects to an MPLS cloud using MPLS ATM tagging. Enters subinterface configuration mode.
Step 2	`Router(config-subif)# ip address address`	Assigns an IP address to the ATM subinterface.
Step 3	`Router(config-subif)# tag-switching atm vp-tunnel vpi`	Specifies an interface or subinterface as a virtual private (VP) tunnel.
Step 4	`Router(config-subif)# tag-switching ip`	Enables label switching of IP packets on the interface.

Configuring the VRF for Each VPN

To configure the VRF for each VPN, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the virtual routing instance by assigning a VRF name.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.
Step 3	`Router(config-vrf)# route-target {import | export | both} route-target-ext-community`	Creates a list of import and export route target communities for the specified VRF.

Configuring a Dedicated PVC

To configure a dedicated PVC for each VPN, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port.subinterface-number point-to-point`	Creates a point-to-point ATM subinterface. Enters subinterface configuration mode.
Step 2	`Router(config-subif)# ip vrf forwarding vrf-name`	Associates a VRF with the ATM point-to-point subinterface.
Step 3	`Router(config-subif)# ip address address`	Assigns an IP address to the ATM subinterface.
Step 4	`Router(config-subif)# pvc [vpi/vci number]`	Configures the PVC on the subinterface. Enters PVC configuration mode.
Step 5	`Router(config-subif-pvc)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type on the ATM PVC.

Configuring BGP to Advertise Networks

To configure BGP to advertise the networks for each VPN, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system`	Configures the internal BGP (iBGP) routing process with the autonomous system number passed along to other iBGP routers.
Step 2	`Router(config-router)# no bgp default ipv4-unicast`	Disables IPv4 BGP routing.
Step 3	`Router(config-router)# neighbor {ip-address | peer-group-name} remote-as as-number`	Configures the neighboring PE router's IP address or iBGP peer group and identifies it to the local autonomous system. The MP-BGP neighbors must use the loopback addresses.
Step 4	`Router(config-router)# neighbor {ip-address | peer-group-name} update-source interface-type`	Allows iBGP sessions to use any operational interface for TCP connections.
Step 5	`Router(config-router)# neighbor {ip-address | peer-group-name} activate`	Activates route exchanges with the global BGP neighbors.
Step 6	`Router(config-router)# address-family ipv4 vrf vrf-name`	Enters address family configuration mode and configures the VRF routing table for BGP routing sessions that use standard IPv4 address prefixes. The *vrf-name* argument specifies the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.

	Command	Purpose
Step 7	`Router(config-router-af)# redistribute protocol`	Redistributes routes from one routing domain into another routing domain. The *protocol* argument is the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **ospf**, **static** [**ip**], or **rip**. The **connected** keyword refers to routes that are established automatically by virtue of having enabled IP on an interface.
Step 8	`Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	`Router(config-router)# address-family vpnv4 [unicast]`	Enters address family configuration mode for configuring BGP routing sessions that use standard Virtual Private Network (VPN) Version 4 address prefixes. (Optional) The **unicast** keyword specifies VPN Version 4 unicast address prefixes.
Step 10	`Router(config-router-af)# neighbor {ip-address | peer-group-name} activate`	Activates route exchanges with the global BGP neighbors.
Step 11	`Router(config-router-af)# neighbor {ip-address | peer-group-name} send-community [both]`	Specifies that a community attribute should be sent to a BGP neighbor. The **both** keyword specifies that both community attributes should be sent.
Step 12	`Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 13	`Router(config-router)# exit`	Exits router configuration mode.
Step 14	`Router(config)# interface atm slot/port.subinterface-number point-to-point`	Creates a point-to-point ATM subinterface. Enters subinterface configuration mode.
Step 15	`Router(config-subif)# atm route-bridged ip`	Enables RBE on the subinterface.

Configuring DHCP Option 82 for RBE

To configure DHCP Option 82 support for RBE connections, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp relay information option`	Enables the system to insert the DHCP relay agent information option in VPN suboptions.
Step 2	`Router(config)# rbe nasip source_interface`	Specifies the IP address of an interface on the DHCP relay agent. This is the interface address that is sent to the DHCP server in the agent remote ID suboption.

Example 3-7 enables DHCP option 82 support on the DHCP relay agent by using the **ip dhcp relay information option** command. The **rbe nasip** command configures the router to forward the IP address for Loopback0 to the DHCP server. The value (in hexadecimal) of the agent remote ID suboption is 010100000B0101814058320 and the value of each field is the following:

- Port Type: 0x01
- Version: 0x01

- Reserved: undefined
- NAS IP address: 0x0B010181 (hexadecimal value of 11.1.1.129)
- NAS Port
  - Interface (slot/module/port): 0x40 (The slot/module/port values are 01 00/0/000.)
  - VPI: 0x58 (hexadecimal value of 88)
  - VCI: 0x320 (hexadecimal value of 800)

Example 3-7 Configuring Option 82 for RBE

```
ip dhcp-server 172.16.1.2
!
ip dhcp relay information option
!
interface Loopback0
  ip address 11.1.1.129 255.255.255.192
!
interface ATM4/0
  no ip address
!
interface ATM4/0.1 point-to-point
  ip unnumbered Loopback0
  ip helper-address 172.16.1.2
  atm route-bridged ip
  pvc 88/800
    encapsulation aal5snap
!
interface Ethernet 5/1
  ip address 172.16.1.1 255.255.0.0
!
router eigrp 100
  network 10.0.0.0
  network 172.16.0.0
!
rbe nasip Loopback0
```

Configuring DHCP Relay Support for MPLS VPN Suboptions

To configure DHCP relay support for MPLS VPN suboptions, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp relay information option vpn`	Enables the system to insert VPN suboptions into the DHCP relay agent information option in forwarded BOOTREQUEST messages to a DHCP server. Sets the gateway address to the outgoing interface toward the DHCP server. The VPN suboptions are also added to the BOOTP broadcast packets when the command is configured.

	Command	Purpose
Step 2	`Router(config)# interface type number`	Specifies an interface and enters interface configuration mode.
Step 3	`Router(config-if)# ip helper-address vrf name [global] address`	Forwards UDP broadcasts, including BOOTP, received on an interface. If the DHCP server resides in a VPN or global space that is different from the VPN, the **vrf** *name* or **global** options allow you to specify the name of the VRF or global space where the DHCP server resides. The **vrf** *name* argument is the virtual routing and forwarding (VRF) instance for the VPN. The **global** argument is the global routing table. The *address* argument is the destination broadcast or host address to be used when forwarding UDP broadcasts. You can configure more than one helper address per interface.

In Example 3-8, the DHCP relay receives a DHCP request on Ethernet interface 0/1 and sends the request to the DHCP server located at IP helper address 10.44.23.7, which is associated with the VRF named *red*.

Example 3-8 Configuring DHCP Relay Support for MPLS VPN Suboptions

```
ip dhcp relay information option vpn
!
interface ethernet 0/1
 ip helper-address vrf red 10.44.23.7
!
```

Specifying a VPN ID

To specify a VPN ID, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Creates a VRF routing table and a CEF forwarding table and enters VRF configuration mode. The *vrf-name* argument is the name you assign to the VRF.
Step 2	`Router(config-vrf)# vpn id oui:vpn-index`	Assigns a VPN ID to the VRF. The *oui***:** argument is an organizationally unique identifier. The IEEE organization assigns this identifier to companies. The OUI is restricted to three octets. The *vpn-index* argument identifies the VPN within the company. This VPN index is restricted to four octets.

Example 3-9 assigns a VPN ID to the VRF named *vpn1*.

Example 3-9 Configuring a VPN ID

```
Router(config)# ip vrf vpn1
Router(config-vrf)# vpn id al:3f6c
Router(config-vrf)# end
```

Configuring and Associating Virtual Private Networks

To add a virtual private network (VPN) service to your MPLS configuration, you perform the following tasks:

- Configure VPNs
- Associate VPNs with a virtual template interface

Configuring Virtual Private Networks

To configure dial-in and dial-out virtual private networks (VPNs), perform the following tasks:

- Enable a VPN tunnel
- Configure VPN tunnel authentication

For more information about configuring virtual private networks, see the "Configuring Virtual Private Networks" chapter in the *Cisco IOS Dial Technologies Configuration Guide, Release 12.2*. This chapter describes the procedures used to configure, verify, monitor, and troubleshoot VPNs and also provides configuration examples.

Associating VPNs with a Virtual Template Interface

After you configure the VPNs, associate each one with a virtual template interface. To do this association, perform the following tasks:

- Creating a VRF Configuration for a VPN, page 3-28
- Associating a VRF Configuration for a VPN with a Virtual Template Interface, page 3-29

Note Do not enable VPN service on the fa0/0/0 management interface. The configuration for this interface is included in the configuration file.

Creating a VRF Configuration for a VPN

To create a VRF configuration for a VPN, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.

	Command	Purpose
Step 3	`Router(config-vrf)# vpn id route-distinguisher`	Associates the VPN with the VRF.
Step 4	`Router(config-vrf)# route-target {import | export | both} route-target-ext-community`	Creates a list of import and export route target communities for the specified VRF.

Example 3-10 Creating a VRF Configuration for a VPN

```
ip vrf common
    rd 100:1000
    vpn id 100:1000
    route-target export 100:1000
    route-target import 100:1000
```

Note For more information about creating VRFs, see the "Configuring Virtual Routing and Forwarding Instances" section on page 3-13.

Associating a VRF Configuration for a VPN with a Virtual Template Interface

After you create a VRF configuration for a VPN, associate the VRF with a virtual template interface. The virtual template interface is used to create and configure a virtual access interface (VAI).

To associate a VRF, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip vrf forwarding vrf-name`	Associates the VRF with the virtual template interface.
Step 3	`Router(config-if)# ip unnumbered type number`	Enables IP without assigning a specific IP address to the interface. The *type* and *number* arguments are the type and number of another interface on which the router has an assigned IP address. The interface cannot be another unnumbered interface.

Example 3-11 Associating a VRF Configuration for a VPN with a Virtual Template Interface

```
interface Virtual-Template1
    ip vrf forwarding common
    ip unnumbered Loopback1
```

Note

- For more information about configuring a virtual template interface, see the "Configuring a Virtual Template Interface" section on page 3-17.
- For more information about creating and associating VRFs, see the "Configuring Virtual Routing and Forwarding Instances" section on page 3-13 and the "Associating VRFs" section on page 3-13.

Configuring RADIUS User Profiles for RADIUS-Based AAA

Use the per VRF AAA feature to partition authentication, authorization, and accounting (AAA) services based on a virtual routing and forwarding (VRF) instance. This feature allows the Cisco 10000 router to communicate directly with the customer RADIUS server without having to go through a RADIUS proxy.

For more information about configuring the per VRF AAA feature on the Cisco 10000 series router, see the "Optional Configuration Tasks for LAC" section on page 5-7.

For more information about configuring your RADIUS server, see your RADIUS documentation.

Verifying VPN Operation

To verify VPN operation, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ip vrf`	Displays the defined VRFs and interfaces.
`Router# show ip vrf [{brief | detail | interfaces}] vrf-name`	Displays information about defined VRFs and associated interfaces.
`Router# show ip route vrf vrf-name`	Displays the IP routing table for a VRF.
`Router# show ip protocols vrf vrf-name`	Displays the routing protocol information for a VRF.
`Router# show ip interface interface-number`	Displays the VRF table associated with an interface.
`Router# show ip bgp vpnv4 all [tags]`	Displays information about all BGPs.
`Router# show tag-switching forwarding vrf vrf-name [prefix mask/length][detail]`	Displays label forwarding entries that correspond to VRF routes advertised by this router.

Configuration Examples for RA to MPLS VPN

This section provides configuration examples for the following configurations:

- PPPoA to MPLS VPN Configuration Example, page 3-31
- PPPoE to MPLS VPN Configuration Example, page 3-34
- RBE to MPLS VPN Configuration Example, page 3-38

PPPoA to MPLS VPN Configuration Example

Example 3-12 shows how to configure the RA to MPLS VPN feature on the Cisco 10000 series router. In this example, one VRF is configured with 300 PPPoA sessions.

Example 3-12 Configuring PPPoA to MPLS VPN

```
!Enables the AAA access control model.
aaa new-model
!
!Configures AAA accounting.
aaa authentication login default none
aaa authentication ppp default local
aaa authorization network default local
aaa session-id common
enable password vermont
!
username vpn1 password 0 vpn1
!
!Configures the vpn1 VRF.
ip vrf vpn1
    rd 10:1
    route-target export 10:1
    route-target import 10:1
!
!Configures the policy map for the default class.
policy-map mypolicy
    class class-default
        police 200000 400000 800000 conform-action transmit exceed-action drop
!
no virtual-template snmp
!
!Sets the size of the small and middle buffers.
buffers small permanent 20000
buffers middle permanent 7000
!
!Defines the general loopback interface used for reachability to the router and as a
!source IP address for sessions (IBGP, TDP, and so on).
interface Loopback0
    ip address 10.1.1.1 255.255.255.255
!
!Creates a loopback interface in the vpn1 VRF. You do this for each customer VRF you IP
!unnumber interfaces to.
interface Loopback1
    ip vrf forwarding vpn1
    ip address 10.16.1.1 255.255.255.255
!
!Configures the management interface. You should not configure VPN over the FastEthernet
!interface.
interface FastEthernet0/0/0
ip address 192.168.16.1 255.255.255.0
no ip proxy-arp
!
!Enables label switching of IP packets on the interface.
interface GigabitEthernet1/0/0
    ip address 172.16.4.1 255.255.0.0
    negotiation auto
    tag-switching ip
!
```

```
interface GigabitEthernet2/0/0
    ip address 172.16.3.1 255.255.0.0
    negotiation auto
    tag-switching ip
!
interface ATM3/0/0
    no ip address
    atm flag s1s0 0
    atm sonet stm-4
    no atm ilmi-keepalive
!
interface ATM4/0/0
    no ip address
    load-interval 30
    no atm pxf queuing
    atm sonet stm-4
    no atm ilmi-keepalive
!
interface ATM4/0/0.1 multipoint
    range pvc 3/32 3/354
        encapsulation aa5mux ppp Virtual-Template1
!
interface ATM6/0/0
    no ip address
    no atm pxf queuing
    no atm ilmi-keepalive
!
interface atm6/0/1
    no ip address
    no atm ilmi-keepalive
!
interface ATM6/0/2
    no ip address
    no atm ilmi-keepalive
!
interface ATM6/0/3
    no ip address
    no atm ilmi-keepalive
!
!Enables label switching of IP packets on the interface.
interface POS7/0/0
    ip address 172.16.1.1 255.255.0.0
    keepalive 30
    tag-switching ip
    crc32
!
interface POS8/0/0
    ip address 172.16.2.1 255.255.0.0
    keepalive 30
    tag-switching ip
    crc32
!
!Configures the virtual template and associates the vpn1 VRF with it.
interface Virtual-Template1
    ip vrf forwarding vpn1
    ip unnumbered Loopback1
    peer default ip address pool vpn1
    ppp max-configure 255
    ppp max-failure 255
    ppp authentication chap
    ppp timeout retry 25
    ppp timeout authentication 20
!
!Configures OSPF to advertise networks.
```

```
router ospf 200
    log-adjacency-changes
    auto-cost reference-bandwidth 10000
    network 10.1.1.1 0.0.0.0 area 40
    network 172.16.0.0 0.255.255.255 area 40
!
!Configures BGP to advertise the networks for each VPN.
router bgp 100
    bgp router-id 10.1.1.1
    no bgp default ipv4-unicast
    bgp cluster-id 671154433
    bgp log-neighbor-changes
    bgp bestpath scan-time 30
    bgp scan-time 30
    neighbor 10.1.1.4 remote-as 100
    neighbor 10.1.1.4 update-source Loopback0
    neighbor 10.1.1.4 activate
!
!Enters address family configuration mode to configure the VRF routing table on BGP.
address-family ipv4 vrf vpn1
redistribute connected
no auto-summary
no synchronization
exit-address-family
!
!Configures MP-IBGP.
address-family vpnv4
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 send-community both
exit-address-family
!
!Specifies the IP local pool to use for the vpn1 VRF address assignment.
ip local pool vpn1 192.168.1.1 192.168.2.67
!
!Enters routing information in the routing table.
ip classless
ip route 192.168.16.0 255.255.255.0 198.168.76.1
no ip http server
ip pim bidir-enable
!
!
no cdp run
!Configures RADIUS accounting. radius-server retransmit is on by default and cannot be
removed.
radius-server retransmit 3
radius-server authorization permit missing Service-Type
call admission limit 90
!
```

PPPoE to MPLS VPN Configuration Example

Example 3-13shows how to configure the RA to MPLS VPN feature with one VRF for PPPoE sessions.

Example 3-13 Configuring PPPoE to MPLS VPN

```
!
!Enables the AAA access control model.
aaa new-model
!
!Configures AAA accounting.
aaa authentication login default none
aaa authentication enable default none
aaa authentication ppp default group radius
aaa authorization config-commands
aaa authorization network default local
aaa session-id common
enable password cisco
!
username pppoe password 0 pppoe
username pppoa password 0 pppoa
username common password 0 common
!
!Preprovisions slots in the Cisco 10000 series router for line cards.
card 1/0 1gigethernet-1
card 2/0 1gigethernet-1
card 3/0 1oc12pos-1
card 4/0 1oc12pos-1
card 5/0 1oc12atm-1
card 6/0 1oc12atm-1
card 7/0 4oc3atm-1
card 8/0 4oc3atm-1
!
!Creates the common VRF.
ip vrf common
 rd 100:1000
 route-target export 100:1000
 route-target import 100:1000
!
!Specifies the VPDN group to be used to establish PPPoE sessions and specifies the maximum
!number of PPPoE sessions to be established over a virtual circuit.
vpdn-group pppoe
 accept-dialin
  protocol pppoe
  virtual-template 1
 pppoe limit per-mac 32000
 pppoe limit per-vc 1
!
no virtual-template snmp
!
!Configures the small buffer.
buffers small permanent 15000
!
vc-class atm vpn
  protocol pppoe
  encapsulation aal5snap
!
!Defines the general loopback interface used for reachability to the router and as a
!source IP address for sessions (IBGP, TDP, and so on).
interface Loopback0
 ip address 10.16.3.1 255.255.255.255
 ip ospf network point-to-point
```

```
!
!Creates a loopback interface in the vpn1 VRF. You do this for each customer VRF you IP
!unnumber interfaces to.
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.24.1.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vpn2
 ip address 10.8.1.2 255.255.255.255
!
!Configures the management interface. You should not configure VPN over the FastEthernet
!interface.
interface FastEthernet0/0/0
 ip address 10.9.100.32 255.0.0.0
 no ip proxy-arp
 full-duplex
!
!Enables label switching of IP packets on the interface.
interface GigabitEthernet1/0/0
 ip address 10.1.10.1 255.255.0.0
 no ip redirects
 load-interval 30
 negotiation auto
 tag-switching ip
!
interface GigabitEthernet2/0/0
 ip address 10.2.10.1 255.255.0.0
 no ip redirects
 load-interval 30
 negotiation auto
 tag-switching ip
!
interface POS3/0/0
 ip address 10.3.10.1 255.255.0.0
 no ip redirects
 ip ospf cost 2
 keepalive 30
 tag-switching ip
 crc 32
 clock source internal
 pos scramble-atm
!
interface POS4/0/0
 ip address 10.4.10.1 255.255.0.0
 no ip redirects
 ip ospf cost 2
 keepalive 30
 tag-switching ip
 crc 32
 clock source internal
 pos scramble-atm
!
interface ATM5/0/0
 no ip address
 load-interval 30
 no atm pxf queuing
 atm clock INTERNAL
 atm sonet stm-4
 no atm ilmi-keepalive
!
interface ATM5/0/0.1000 multipoint
 range pvc 2/32 2/63
 !
```

```
 class-int vpn
!
interface ATM6/0/0
 no ip address
 load-interval 30
 no atm pxf queuing
 atm clock INTERNAL
 atm sonet stm-4
 no atm ilmi-keepalive
!
interface ATM6/0/0.1000 multipoint
 range pvc 2/32 2/63
  encapsulation aal5snap
  protocol pppoe
 !
 class-int vpn
!
interface ATM7/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM7/0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM7/0/2
 no ip address
 no atm ilmi-keepalive
!
interface ATM7/0/3
 no ip address
 no atm ilmi-keepalive
!
interface ATM8/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM8/0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM8/0/2
 no ip address
 no atm ilmi-keepalive
!
interface ATM8/0/3
 no ip address
 no atm ilmi-keepalive
!
interface ATM8/0/3.100 multipoint
 range pvc 2/32 2/42
  encapsulation aal5snap
  protocol pppoe
 !
!Associates the common VRF with the interface.
interface ATM8/0/3.101 point-to-point
 ip vrf forwarding common
 ip address 10.22.10.1 255.255.255.0
 pvc 3/32
  encapsulation aal5snap
 !
```

```
!Defines the virtual template and associates the common VRF with it.
interface Virtual-Template1
 ip vrf forwarding common
 ip unnumbered Loopback1
 peer default ip address pool common
 ppp authentication chap
!
!Configures OSPF to advertise the networks.
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 10.16.3.1 0.0.0.0 area 0
 network 10.1.0.0 0.0.255.255 area 0
 network 10.2.0.0 0.0.255.255 area 0
 network 10.3.0.0 0.0.255.255 area 0
 network 10.4.0.0 0.0.255.255 area 0
!
router rip
 version 2
 !
!Enters address family configuration mode to configure the VRF for PE to CE routing
!sessions.
 address-family ipv4 vrf common
 version 2
network 10.0.0.0
 no auto-summary
 exit-address-family
!
!Configures BGP to advertise the networks for the VPN.
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 172.16.1.4 remote-as 100
 neighbor 172.16.1.4 activate
 !
!Enters address family configuration mode to configure the common VRF for PE to CE routing
!sessions.
address-family ipv4 vrf common
 no auto-summary
 no synchronization
 aggregate-address 2.10.0.0 255.255.0.0 summary-only
 exit-address-family
 !
 address-family vpnv4
 neighbor 172.16.1.4 activate
 neighbor 172.16.1.4 send-community both
 exit-address-family
!
!Specifies the IP local pool to use for the VRF address assignment.
ip local pool common 2.10.1.1 2.10.126.0
ip classless
!Enters routing information in the routing table for the VRF.
ip route 20.0.0.0 255.0.0.0 FastEthernet0/0/0 20.9.0.1
ip route vrf common 10.22.0.0 255.255.0.0 Null0
ip route vrf common 10.30.0.0 255.255.0.0 2.1.1.1 3
ip route vrf common 10.32.0.0 255.255.0.0 2.2.151.1 2
ip route vrf common 10.33.0.0 255.255.0.0 2.3.101.1 2
no ip http server
ip pim bidir-enable
!
no cdp run
!
```

```
!Specifies the RADIUS host and configures RADIUS accounting. radius-server retransmit is
!on by default and cannot be removed.
radius-server host 10.19.100.150 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key test
radius-server authorization permit missing Service-Type
radius-server vsa send authentication
call admission limit 90
!
```

RBE to MPLS VPN Configuration Example

Example 3-14 shows how to configure RBE on ATM interfaces, creates and associates two VRFs named *CustomerA* and *CustomerB*, and configures DHCP Option 82 support for RBE connections.

Example 3-14 Configuring RBE to MPLS VPN

```
ip vrf CustomerA
rd 100:100
route-target export  100:100
route-target import  100:100
!
ip vrf CustomerB
rd 101:101
route-target export  101:101
route-target import  101:101
!
interface int g1/0/0
ip address 192.168.1.1 255.255.255.0
tag-switching ip
!
interface loopback0
! BGP update source
ip address 10.100.10.1 255.255.255.255
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
redistribute connected
!
interface loopback 1
! description for VRF CustomerA
ip address 10.101.10.1 255.255.255.255
ip vrf forwarding CustomerA
!
interface loopback 2
! description for VRF CustomerB
ip address 10.102.20.1 255.255.255.255
ip vrf forwarding CustomerB
!
ip dhcp relay information option
ip dhcp relay information option vpn
!
interface atm7/0/0
no atm pxf queuing
!
interface atm7/0/0.1 point-to-point
ip vrf forwarding CustomerA
ip unnumbered loopback1
 ip helper-address vrf CustomerA 192.168.2.1
atm route ip
range pvc 101/32 101/2031
```

```
encapsulation aal5snap
!
interface atm8/0/0
no atm pxf queuing
!
interface atm8/0/0.1 point-to-point
ip vrf forwarding CustomerB
ip unnumbered loopback2
 ip helper-address vrf CustomerB 192.168.3.1
atm route ip
range pvc 102/32 102/2031
encapsulation aal5snap
!
router bgp 1
no synchronization
redistribute connected
neighbor 192.168.1.2 remote-as 1
neighbor 192.168.1.2 update source loopback0
neighbor 192.168.1.2 activate
no auto-summary
!
address-family ipv4 vrf CustomerA
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf CustomerB
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 192.168.1.2 activate
neighbor 192.168.1.2 send-community extended
no auto-summary
exit-address-family
```

Monitoring and Maintaining an MPLS Configuration

To monitor and maintain an MPLS configuration, perform the following verification tasks:

- Verifying the Routing Protocol Is Running, page 3-40
- Verifying MPLS, page 3-40
- Verifying Connections Between Neighbors, page 3-40
- Verifying Label Distribution, page 3-41
- Verifying Label Bindings, page 3-42
- Verifying Labels Are Set, page 3-43

For more information, see the “Troubleshooting Tag and MPLS Switching Connections” chapter in the *ATM and Layer 3 Switch Router Troubleshooting Guide, Cisco IOS Release 12.1(13)E1*.

Verifying the Routing Protocol Is Running

To verify that the routing protocol is running, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ip protocols`	Displays the parameters and current state of the active routing protocol process. Ensure that the protocol routes for the MPLS network and all neighbors are present.
`Router# show ip route`	Displays the current state of the routing table. Ensure that all routers and routes are present.

Verifying MPLS

To verify MPLS, enter the following command in privileged EXEC mode:

Command	Purpose
`Router# show mpls interfaces`	Displays information about the interfaces that have been configured for label switching. Use this command to verify that MPLS is globally enabled and that a label distribution protocol is running on the requested interfaces.

Example 3-15 show mpls interfaces

```
Router# show mpls interfaces

InterfaceIPTunnelOperational
(...)
Serial0/1.1Yes (tdp)YesYes
Serial0/1.2YesYesNo
Serial0/1.3Yes (tdp)YesYes
(...)
```

The fields in this example indicate the following:

- IP field—Indicates that MPLS IP is configured for an interface. The label distribution protocol (LDP) appears in parentheses to the right of the IP status. The LDP is either Tag Distribution Protocol (TDP) as defined in the Cisco Tag Switching architecture, or LDP as defined by IETF in RFC 3036.
- Tunnel field—Indicates the capacity of traffic engineering on the interface.
- Operational field—Indicates the status of the LDP. In the above example, the Operational field indicates down on Serial 0/1.2 because the interface is down.

Verifying Connections Between Neighbors

An unlabeled connection must exist between each pair of neighboring routers. The routing protocol and the label distribution protocol use the unlabeled connection to build the outing table and the Label Forwarding Information Base (LFIB).

To verify the connections between neighbors, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **ping** [*protocol* | **tag**] {*host-name* | *system-address*}	Verifies basic network connectivity between neighbors.
Router# **ping vrf** *vrf-name system-address*	Verifies connectivity to the VRF specified.
Router# **debug mpls packet**	Verifies that MPLS labels are set.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Example 3-16 ping

```
Router# ping 10.10.10.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

Example 3-17 ping vrf

```
Router# ping vrf vrf-1 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4/ ms
```

Verifying Label Distribution

To verify label distribution, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show mpls forwarding-table**	Displays the discovered neighbors. The Local Tag field displays the label assigned by the router.
Router# **show tag-switching tdp discovery**	Displays the status of the LDP discovery process.

Example 3-18 show mpls forwarding-table Command

```
Router# show mpls forwarding-table

LocalOutgoingPrefixBytes tagOutgoingNext Hop
tag tag or VCor Tunnel Idswitchedinterface
16  Untagged10.1.0.0/160AT9/0/010.4.4.2
17  Untagged10.0.0.0/80AT9/0/010.4.4.2
18  Untagged192.168.0.0/160AT9/0/110.6.6.2
19  Pop tag192.168.2.1/32624Fal1/0/0172.16.0.1
20  Pop tag192.168.2.2/320Fal1/0/1172.16.0.18
```

In Example 3-19, TDP is used to bind labels with routes. If label distribution protocol is running correctly, it assigns one label per forwarding equivalent class. If any of the presumed neighbors is missing and cannot be pinged, a connectivity problem exists and the label distribution protocol cannot run.

Example 3-19 show tag-switching tdp discovery Command

```
Router# show tag-switching tdp discovery

Local TDP Identifier:
    10.10.10.3:0
Discovery Sources:
    Interfaces:
        Serial0/1.1 (tdp): xmit/recv
           TDP Id: 10.10.10.1:0
        Serial0/1.2 (tdp): xmit/recv
           TDP Id: 10.10.10.2:0
        Serial0/1.3 (tdp): xmit/recv
           TDP Id: 10.10.10.6:0
```

Note The neighbor relationship is not established when the router ID for the label distribution protocol cannot be reached from the global routing table.

Verifying Label Bindings

To verify label bindings, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show mpls ip bindings`	Displays the labels assigned to each destination.
`Router# show mpls tag-switching forwarding-table {ip-address | prefix} detail`	Displays the different routes and the labels associated with them.

Example 3-20 show mpls ip bindings Command

```
Router# show mpls ip binding

    10.4.4.0/24
        in label:imp-null
        out label:imp-nulllsr: 172.16.1.18:0
    10.6.6.0/24
        in label:imp-null
        out label:imp-nulllsr: 172.16.1.18:0
```

```
10.0.0.0/8
    in label:17
10.18.0.0/8
    out label:16
172.16.1.0/30
    in label:imp-null
    out label:imp-nulllsr: 192.168.1.1:0
    out label:20lsr: 172.16.1.18:0
172.16.1.16/30
    in label:imp-null
    out label:16lsr: 192.168.1.1:0
    out label:imp-nulllsr: 172.16.1.18:0
```

Verifying Labels Are Set

To verify that the labels are set, enter the following command in privileged EXEC mode:

Command	Purpose
`Router# traceroute address`	Displays the route to the specified address and the labels set for the interfaces.

Example 3-21 traceroute Command

```
Router# traceroute 10.10.10.4

Type escape sequence to abort.
Tracing the route to 10.10.10.4
    1 10.1.1.21 [MPLS: Label 25 Exp 0] 296 msec 256 msec 244 msec
    2 10.1.1.5 [MPLS: Label 22 Exp 0] 212 msec 392 msec 352 msec
    3 10.1.1.14 436 msec * 268 msec
```

Monitoring and Maintaining the MPLS VPN

To monitor and maintain an MPLS VPN configuration, perform the following verification tasks:

- Verifying VRF Configurations, page 3-44
- Verifying the Routing Table, page 3-44
- Verifying the PE to PE Routing Protocols, page 3-45
- Verifying the PE to CE Routing Protocol, page 3-46
- Verifying the MPLS VPN Labels, page 3-46
- Testing the VRF, page 3-46

Note Before you establish an MPLS VPN, verify the connections between PE routers by using the **ping** command.

Verifying VRF Configurations

To verify VRF configurations, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show ip vrf**	Displays a summary of all VRFs present on the current router and their associated route distinguishers and interfaces. Use this command to verify the names and configuration of each VRF and the route distinguisher configuration at each PE router.
Router# **show ip vrf interfaces**	Displays the VRFs present on the router and the associated interfaces.
Router# **show ip vrf detail** *vrf-name*	Displays detailed information about the VRF you specify. Use this command to determine if the global routing table contains all connected addresses, if the exported routing attributes of a VRF on a PE router are the imported routing attributes of the VRF on another PE router, and to determine the status and IP addresses of interfaces.

Example 3-22 show ip vrf interfaces Command

```
Route# show ip vrf interfaces

InterfaceIP-AddressVRFProtocol
Loopback101100.0.6.1vrf-1up
Loopback111200.1.6.1vrf-2up
```

Example 3-23 show ip vrf detail vrf-name

```
Router# show ip vrf detail vrf-1

VRF vrf-1; default RD 100:101
    Interfaces:
        Loopback101Loopback111
    Connected addresses are not in global routing table
    Export VPN route-target communities
        RT:100:1001
    Import VPN route-target communities
        RT:100:1001
    No import route-map
    No export route-map
```

Verifying the Routing Table

To verify the routing table for VRFs, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show ip route vrf** *vrf-name*	Displays MPLS VPN connections in the routing table.
Router# **show ip route vrf** *vrf-name system-address*	Displays routing table information for the specified address.

Verifying the PE to PE Routing Protocols

Border Gateway Protocol (BGP) is used for routing sessions between PE routers. To verify PE to PE routing sessions, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show ip bgp neighbors**	Displays detailed information on the BGP and TCP connections to individual neighbors.
Router# **show ip bgp vpnv4 all**	Shows the VPN address information from the BGP table.
Router# **show ip bgp vpnv4 vrf** *vrf-name*	Displays network layer reachability information associated with the specified VRF.
Router# **show ip bgp vpnv4 vrf** *vrf-name ip-address*	Displays network layer reachability information associated with the specified VRF and a specific connection.

Example 3-24 show ip bgp vpnv4 all Command

```
Router# show ip bgp vpnv4 all

BGP table version is 17, local router ID is 192.168.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Route Distinguisher: 100:1 (default for vrf vrf-1)
*>i10.1.1.0/24192.168.1.101000101?
*>i172.16.1.100/30192.168.1.101000?
*> 172.16.1.116/300.0.0.0032768?
*>i172.16.42.0/24192.168.1.101000101?
*>i192.168.2.1/32192.168.1.101000101i
*> 192.168.5.1/32172.16.1.11800202i
Route Distinguisher: 200:1 (default for vrf vrf-2)
*>i172.16.2.100/30192.168.1.101000?
*> 172.16.2.116/300.0.0.0032768?
```

Example 3-25 show ip bgp vpnv4 vrf vrf-name ip-address Command

```
Router# show ip bgp vpnv4 vrf vrf-1 172.16.2.116

BGP routing table entry for 200:1:172.16.2.116/30, version 7
Paths: (1 available, best #1, table vrf-1)
    Advertised to non peer-group peers:
    192.168.1.1
    Local
        0.0.0.0 from 0.0.0.0 (102.168.1.2)
           Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced
           Extended Community: RT:200:1
```

Verifying the PE to CE Routing Protocol

If the CE router uses a routing protocol other than BGP (for example, RIP or OSPF), enter any of the following commands in privileged EXEC mode to verify the PE to CE routing sessions:

Command	Purpose
Router# **show ip rip database vrf** *vrf-name*	Displays summary address entries in the Routing Information Protocol (RIP) routing database for the specified VRF.
Router# **show ip ospf** [*process-id [area-id*]] **database**	Displays lists of information related to the OSPF database for a specific router.

Note The **show ip rip database vrf** and **show ip ospf** commands are useful for verifying the routing table from the CE router side of the connection and for determining if neighbors are missing from the routing table.

Verifying the MPLS VPN Labels

An MPLS VPN uses a transport label to identify the VRF and another label to identify the backbone. To verify the MPLS VPN labels, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **traceroute vrf** *vrf-name ip-address*	Displays the transport addresses for the specified VRF. Ensure that the interfaces displayed are the correct cross-connect addresses.
Router# **show ip bgp vpnv4 all tags**	Displays the labels for a particular VRF.

Note The **traceroute vrf** command works with an MPLS-aware traceroute, and only if the backbone ATM switch routers are configured to propagate and generate IP Time to Live (TTL) information.

Example 3-26 traceroute vrf Command

```
Router# traceroute vrf vrf-1 192.168.1.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

    1 10.0.1.17 4 msec 0 msec 4 msec
    2 10.0.1.101 0 msec 0 msec 0 msec
    3 10.0.1.102 4 msec * 0 msec
```

Testing the VRF

To test the VRF to ensure that it is working properly, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **ping** [*protocol* | **tag**] {*host-name* | *system-address*}	Verifies basic network connectivity between neighbors.
Router# **ping vrf** *vrf-name system-address*	Tests network connectivity of the specified VRF from the PE router.

Example 3-27 ping vrf vrf-name system-address Command

```
Router# ping vrf vrf-1 192.168.6.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 176/264/576 ms
```

Monitoring and Maintaining PPPoX to MPLS VPN

To monitor and maintain PPPoX to MPLS VPN environments, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show atm pvc ppp**	Displays all ATM PVCs and PPPoA traffic information. **Note** This command applies only to PPPoA sessions.
Router# **show int virtual-access** *virtual access interface #*	Displays status, traffic data, and configuration information about a specified virtual access interface.
Router# **show ip route vrf** *vrf-name*	Displays the IP routing table associated with a VRF.
Router# **show ip local pool**	Displays statistics for any defined IP address pools.
Router# **show vpdn session** [**all**]	Displays information about active L2TP tunnel and message identifiers in a virtual private dialup network (VPDN). **Note** This command applies to PPPoE sessions.
Router# **show vpdn tunnel**	Displays information about active L2TP tunnel and message identifiers in a VPDN. **Note** This command applies to PPPoE sessions.
Router# **debug aaa authentication**	Displays information about AAA authentication.
Router# **debug aaa authorization**	Displays information about AAA authorization.
Router# **debug ip peer**	Displays address activity and contains additional output when pool groups are defined.
Router# **debug ppp negotiation**	Displays PPP packets transmitted during PPP startup where PPP options are negotiated.
Router# **debug ppp authentication**	Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.

Command	Purpose
`Router# debug radius`	Displays information associated with the Remote Authentication Dial-In User (RADIUS) server.
`Router# debug vpdn pppoe-events`	Displays PPPoE protocol errors that prevent a session from being established or errors that cause an established session to be closed.
`Router# debug vtemplate`	Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time the virtual access interface comes down when the call ends.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Note For more information, see the "Troubleshooting DSL Access to MPLS VPN Integration" chapter in the *Troubleshooting Cisco Remote Access to MPLS VPN Integration, Release 2.0*.

Monitoring and Maintaining RBE to MPLS VPN

To monitor and maintain RBE to MPLS VPN environments, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show atm map`	Displays the list of all configured ATM static maps to remote hosts on an ATM network and on ATM bundle maps. **Note** This command enables you to confirm the configuration of the map statements in a static map. This command is useful when an encapsulation failure occurs on a packet because a Layer 3 address could not be mapped to a corresponding Layer 2 address.
`Router# show atm vc`	Displays all ATM PVCs, SVCs, and traffic information.
`Router# show interfaces atm` *interface*	Displays information about the ATM interface.
`Router# show ip arp vrf` *name*	Displays the Address Resolution Protocol (ARP) cache associated with a VRF.
`Router# show ip route vrf` *name*	Displays the IP routing table associated with a VRF.

Command	Purpose
`Router# debug ip packet`	Displays general IP debugging information and IP security option (IPSO) security transactions. **Note** This command is useful if the RFC 1483 PVC does not connect.
`Router# debug ip dhcp`	Displays information about DHCP client activities and the status of DHCP packets.
`Router# debug ip dhcp server events`	Reports server events, such as address assignments and database updates.
`Router# debug ip dhcp server packet`	Decodes DHCP receptions and transmissions.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Note For more information, see the "Troubleshooting DSL Access to MPLS VPN Integration" chapter in the *Troubleshooting Cisco Remote Access to MPLS VPN Integration, Release 2.0*.

CHAPTER 4

Configuring Multiprotocol Label Switching

Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables service providers to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today's networks.

This chapter describes the following MPLS-related features:

For more information about MPLS, see Chapter 3, "Configuring Remote Access to MPLS VPN" and see the *Multiprotocol Label Switching on Cisco Routers, Release 12.1(3)T feature module*.

BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

Load sharing is a concept that allows the Cisco 10000 series router to take advantage of multiple best paths to a given destination. The paths are derived either statically or with dynamic protocols such as RIP, BGP, OSPF, and IGRP. The best path algorithm decides which is the best path to install in the IP routing table and to use for forwarding traffic.

The BGP Multipath Load Sharing for eBGP and iBGP in an MPLS VPN feature allows you to configure multipath load sharing with both external Border Gateway Protocol (eBGP) and internal BGP (iBGP) paths in BGP networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). BGP Multipath Load Sharing provides improved load sharing deployment and service offering capabilities and is useful for multihomed autonomous systems and provider edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

BGP installs up to the maximum number of paths allowed (configured using the **maximum-paths** command). BGP uses the best path algorithm to select one multipath as the best path, insert the best path into the routing information base (RIB), and advertise the best path to BGP peers. Other multipaths may be inserted into the RIB, but only one path is selected as the best path.

Note The maximum number of configurable paths on the PRE2 is 6.

Cisco Express Forwarding (CEF) uses the multipaths to perform load sharing, which can be performed on a per-packet or per-source/destination pair basis. By default, the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature performs unequal cost load sharing by selecting BGP paths that do not have an equal cost of the Interior Gateway Protocol (IGP). To enable the feature, configure the router with MPLS VPNs that contain virtual routing and forwarding instances (VRFs) that import both eBGP and iBGP paths. You can configure the number of multipaths separately for each VRF.

Note The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature operates within the configuration parameters of the existing outbound routing policy.

The BGP Multipath Load Sharing for eBGP and iBGP in an MPLS VPN feature is described in the following topics:

Feature History for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2
12.2(33)SB	The IGP convergence acceleration feature was added on Cisco 10000 series router.	PRE3 and PRE4

Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature has the following restrictions:

- The Cisco 10000 series router supports recursive load sharing, but with the following restriction.

  In recursive load sharing, the information required to forward a packet requires at least 2 lookups. The first lookup determines which provider edge (PE) router is used to reach the final destination. The second lookup determines how to reach the PE router (from first lookup).

  When you configure MPLS VPN, CEF uses recursive load sharing. The first lookup provides the VPN label, the second lookup provides the IGP label. When PXF forwards a packet, it does only 1 lookup which provides both a VPN and an IGP label; 2 lookups in CEF are combined into 1. The restriction for recursive load sharing when PXF forwards a packet is as follows.

  When there are multiple IGP paths between a Cisco 10000 Series PE router to a provider router (P), only per-tag load sharing is supported. That is, PXF is programmed with only one of the paths and this one path is chosen in a round-robin fashion. Because the path is chosen at prefix setup time, it is not possible to predict which path will be selected for which prefix. The path selected depends on the order in which the prefixes are configured in the routing table. The bandwidths of the IGP paths are not considered in the path selection.

Note From Cisco IOS Release 12.2(33)SB onwards, Cisco 10000 series routers support IGP and VPN load balancing for MPLS VPN scenarios on PRE3 and PRE4 engines. This allows faster failover of IGP routes during load balancing.

- When the routing table contains multiple iBGP paths, a route reflector advertises only one of the paths (one next hop). If a router is behind a route reflector, all routers that are connected to multihomed sites are not advertised unless separate VRFs with different route distinguishers (RDs) are configured for each VRF.
- Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses additional memory. We recommend not using this feature on a router with a low amount of available memory and especially when the router is carrying a full Internet routing table.

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature has the following requirements:

- MPLS VRFs must be configured before load sharing with both eBGP and iBGP routes can be configured.

Configuration Tasks for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

To configure the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature, perform the following configuration tasks:

Configuring Multipath Load Sharing for eBGP and iBGP

To configure iBGP and eBGP routes for multipath load sharing, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp as-number`	Configures the router to run a BGP process and enters router configuration mode.
Step 2	`Router(config-router)# address-family ipv4 vrf vrf-name`	Configures a VRF instance for an IPv4 session and enters address family configuration mode.
Step 3	`Router(config-router-af)# maximum-paths eibgp number-of-paths`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. The maximum number of configurable paths on the PRE2 is 6. **Note** You must configure the **maximum-paths eibgp** command in address family IPv4 VRF configuration mode. You cannot configure the command in any other address family configuration mode.

Example 4-1 shows how to configure a VRF named *main* for an IPv4 session and configures a router to select 4 eBGP or iBGP paths as multipaths in address family configuration mode.

Example 4-1 Configuring eBGP and iBGP Multipath Load Sharing

```
Router(config)# router bgp 50
Router(config-router)3 address-family ipv4 vrf main
Router(config-router-af)# maximum-paths eibgp 4
```

Verifying Multipath Load Sharing for Both eBGP and iBGP

To verify that iBGP and eBGP routes are configured for load sharing, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ip bgp vpnv4 all`	Displays all available VPNv4 information from the BGP database.
`Router# show ip route vrf vrf-name`	Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance.

Configuration Examples for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN

This section provides the following configuration examples:

eBGP and iBGP Multipath Load Sharing Configuration Example

Example 4-2 configures a router to select six eBGP or iBGP paths as multipaths in address family configuration mode:

Example 4-2 Configuring eBGP and iBGP Multipath Load Sharing

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf vrf-1
Router(config-router-af)# maximum-paths eibgp 6
```

Verifying eBGP and iBGP Multipath Load Sharing

Example 4-3 shows sample output displayed when you enter the **show ip bgp vpnv4** command. The third line of output (`Multipath:eiBGP`) indicates that multipath load sharing is on.

Example 4-3 Verifying eBGP and iBGP Multipath Load Sharing

```
Router# show ip bgp vpnv4 all 10.22.22.0

BGP routing table entry for 10:1:22.22.22.0/24, version 19
Paths: (5 available, best #5)
Multipath:eiBGP
    Advertised to non peer-group peers:
    10.0.0.2 10.0.0.3 10.0.0.4 10.0.0.5
    22
        10.0.0.0 (metric 20) from 10.0.0.4 (10.0.0.4)
            Origin IGP, metric 0, localpref 100, valid, internal, multipath
            Extended Community:0x0:0:0 RT:100:1 0x0:0:0
            Originator:10.0.0.2, Cluster list:10.0.0.4
    22
        10.0.0.2 (metric 20) from 10.0.0.5 (10.0.0.5)
            Origin IGP, metric 0, localpref 100, valid, internal, multipath
            Extended Community:0x0:0:0 RT:100:1 0x0:0:0
            Originator:10.0.0.2, Cluster list:10.0.0.5
    22
        10.0.0.2 (metric 20) from 10.0.0.2 (10.0.0.2)
            Origin IGP, metric 0, localpref 100, valid, internal, multipath
            Extended Community:RT:100:1 0x0:0:0
    22
        10.0.0.2 (metric 20) from 10.0.0.3 (10.0.0.3)
            Origin IGP, metric 0, localpref 100, valid, internal, multipath
            Extended Community:0x0:0:0 RT:100:1 0x0:0:0
            Originator:10.0.0.2, Cluster list:10.0.0.3
    22
        10.1.1.12 from 10.1.1.12 (10.22.22.12)
            Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
            Extended Community:RT:100:1
```

Monitoring and Maintaining BGP Multipath Load Sharing for eBGP and iBGP

To display eBGP and iBGP multipath load sharing information, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ip bgp all neighbors`	Displays information about the TCP and BGP connections to neighbors.
`Router# show ip bgp vpnv4 all ip-prefix/length`	Displays attributes and multipaths for a network in an MPLS VPN. The *ip-prefix* option is an IP prefix address (in dotted decimal format) and the *length* option is the length of the mask (0 to 32). You must include the slash mark when you use the *length* option.
`Router# show ip bgp vpnv4 all labels`	Displays incoming and outgoing BGP labels for each NLRI prefix.
`Router# show ip bgp vpnv4 rd route-distinguisher`	Displays Network Layer Reachability Information (NLRI) prefixes that have a matching route distinguisher.
`Router# show ip bgp vpnv4 all summary`	Displays BGP neighbor status.
`Router# show ip bgp vpnv4 vrf vrf-name`	Displays NLRI prefixes associated with the named virtual routing and forwarding instance (VRF).
`Router# show ip route vrf vrf-name ip-prefix`	Displays routing information for a network in an MPLS VPN.

IPv6 VPN over MPLS (6VPE)

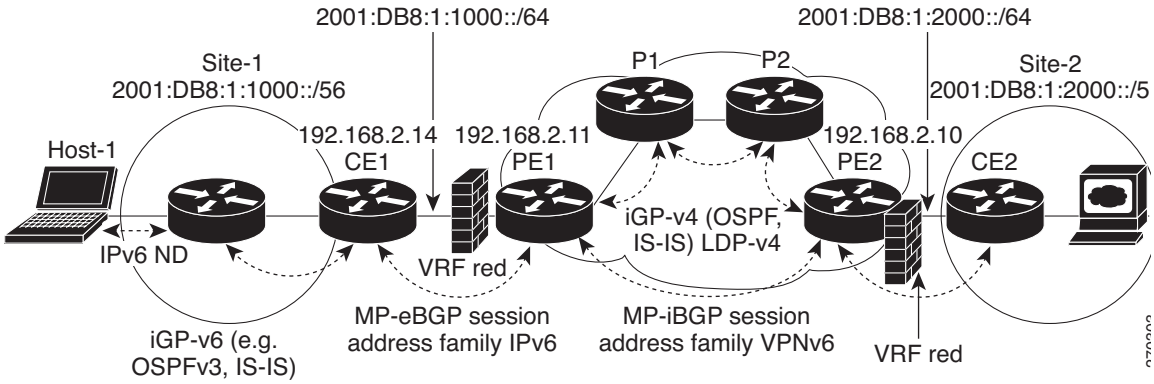
Multiprotocol BGP is the centerpiece of the MPLS IPv6 VPN architecture in both IPv4 and IPv6. It is used to distribute IPv6 routes over the service provider backbone, with the same set of mechanisms to work with overlapping addresses, redistribution policies, and scalability issues. The 6VPE feature is the IOS implementation as described in RFC 4659.

Although IPv6 should not have overlapping address space through the use of global IPv6 Unicast prefix—RFC 3587—or Unique IPv6 Local Addressing—RFC 4193. A network layer reachability information (NLRI) 3-tuple format, which contains length, IPv6 prefix, and label, is defined to distribute these routes using multiprotocol BGP. The extended community attribute—the route target—is used to control redistribution of routing information by tagging exported routes and filtering imported ones.

For scalability, route reflectors can be used to concentrate routing paths and avoid a full PE mesh. Similar to IPv4, BGP features in IPv6, such as route refresh, automatic route filtering, and outbound route filtering, help reduce the number of routes held in each PE.

Figure 4-1 illustrates the important aspects of the IPv6 VPN architecture.

Figure 4-1 Simple IPv6 VPN architecture



The IPv6 VPN over MPLS (6VPE) feature is described in the following topics:

Feature History for IPv6 VPN over MPLS (6VPE)

Cisco IOS Release	Description	Required PRE
12.2(33)SB	This feature has been introduced on Cisco 10000 series routers.	PRE2, PRE3 and PRE4

Prerequisites for Implementing IPv6 VPN over MPLS

The following Cisco IOS services must be running on the network before you configure IPv6 VPN operation:

- MPLS in provider backbone routers
- MPLS with VPN code in provider routers with VPN PE routers
- BGP in all routers providing a VPN service
- Cisco Express Forwarding switching in every MPLS-enabled router
- The **ipv6 unicast-routing** command is enabled on VPN PE routers

Restrictions for Implementing IPv6 VPN over MPLS

The 6VPE feature has the following restrictions:

- 6VPE is supported by an MPLS IPv4-signaled core. An MPLS IPv6-signaled core is not supported.

- The maximum number of IPv6 VRF's that can be supported are 2038, including the global routing instance. However, in 2038 VRF's, only 1200 eBGP sessions are supported and the remaining VRF's are to be static routed.
- The minimum number of routes supported across all IPv6 VRFs, including the global routing instance, is 50K, yielding approximately 24 routes/VRF.

Configuration tasks for Implementing IPv6 VPN over MPLS (6VPE)

Tip The 12.2(33)SRB release introduced the Implementing IPv6 VPN over MPLS (6VPE) feature, you can see the Implementing IPv6 Addressing and Basic Connectivity chapter in the *Cisco IOS IPv6 Configuration Library* as a basic reference to the feature at: http://www.cisco.com/en/US/docs/ios/12_2t/ipv6/v6addres.html

For the configuration tasks in Implementing IPv6VPN over MPLS (6VPE), see the Implementing IPv6 VPN over MPLS (6VPE) chapter in the *Cisco IOS IPv6 Configuration Guide, Release 12.2SR* for the following features at:

http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-ov_mpls_6vpe.html

- Configuring a Virtual Routing and Forwarding Instance for IPv6
- Binding a VRF to an Interface
- Configuring a Static Route for PE-to-CE-Routing
- Configuring eBGP PE-to-CE Routing Sessions
- Configuring the IPv6 VPN Address Family for iBGP
- Configuring Route Reflectors for Improved Scalability
- Configuring Internet Access

Note Cisco 10000 does not support the **mpls ipv6 vrf** command that has been listed as one of the steps to configure VRF for IPv6.

The IPv6VPN over MPLS (6VPE) feature also supports the configuration of the following features on Cisco 10000 series router:

- BGP Features
- IPv6 Internet Access
- VRF-aware Router Applications
- VRF-Lite
- QoS features

BGP Features

The following features are supported on Cisco 10000 series router by the IPv6VPN over MPLS (6VPE) feature:

- Site of Origin (SoO)

SoO is used to prevent routing loops in the case of a Dual-homed CE. The 6VPE feature supports the SoO Attribute for control of IPv6 VPN routes exactly in the same way as it is currently supported for IPv4 VPNs.

For information on configuring this feature, see the How to Configure EIGRP MPLS VPN PE-CE Site of Origin (SoO) Support section in the *EIGRP MPLS VPN PE-CE Site of Origin (SoO)* guide at:

http://www.cisco.com/en/US/docs/ios/12_4/ip_route/configuration/guide/h_mvesoo_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1048097

- ASN Override

  If a global ASN is specified, BGP automatically replaces the ASN with a unique number to ensure the site is uniquely identified. The 6VPE feature supports the ASN Override BGP feature via the use of the 'as-override' keyword in the same way as the feature is currently supported by IPv4 VPNs.

- Allow-AS-in

  A BGP speaker normally ignores a received update that contains its own ASN in **AS_PATH** attribute. This check must be omitted in the case of hub-and-spoke topology. The 6VPE feature supports the Allow-AS-in BGP feature via the use of the **allowas-in** keyword in the same way as the feature is currently supported by IPv4 VPNs.

- BGP Prefix List Filtering

  The 6VPE feature supports the ability to filter MP-BGP IPv6 advertisement based on configured IPv6 Prefixes.

  For information on configuring this feature, see the Configuring BGP Filtering Using Prefix Lists section in the Configuring BGP chapter of the *Cisco IOS IP Configuration Guide, Release 12.2* guide at:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html#wp1001470

- BGP AS Path Filtering

  The 6VPE feature supports the ability to filter MP-BGP IPv6 advertisement based on configured AS Paths.

  For information on configuring this feature, see the Configuring BGP Path Filtering by Neighbor section in the Configuring BGP chapter of the *Cisco IOS IP Configuration Guide, Release 12.2* guide at:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html#wp1001644

- BGP Max Prefix

  The 6VPE feature supports an upper limit on the number of BGP routes that has been learnt from a given CE. The 6VPE feature also supports the Max Prefix BGP feature via the use of the 'maximum-prefix' keyword in the same way as the feature is currently supported by IPv4 VPNs.

  For information on configuring this feature, see the Configuring BGP section in the Configuring BGP chapter of the *Cisco IOS IP Configuration Guide, Release 12.2* guide at:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html

- BGP Route Refresh

  Using BGP Route Refresh, a MP-BGP speaker (PE and/or CE) can request another BGP Speaker to re-send their MP-BGP updates.

  For information on configuring this feature, see the Configuring BGP section in the Configuring BGP chapter of the *Cisco IOS IP Configuration Guide, Release 12.2* guide at:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html

- Route Target Rewrite at AS Boundary

  The 6VPE feature supports the Route Target Rewrite at AS Boundary feature in the same way as the feature is currently supported by IPv4 VPNs.

  For information on configuring this feature, see the Inter-AS RT-Rewrite section in the Spanning Multiple Autonomous Systems chapter of the *Cisco IP Solution Center MPLS VPN User Guide, 5.0* guide at:

  http://www.cisco.com/en/US/docs/net_mgmt/ip_solution_center/5.0.1/mpls_vpn/user/guide/multauto.html#wp631364

- BGP Multipath

  The 6VPE feature supports eBGP Multipath, iBGP Multipath, eiBGP Multipath and the DMZ-link-bandwidth based load-balancing for IPv6 VPNs in the VPN-IPv6 address family, in the same way as they are currently supported by IPv4 VPNs in the VPN-IPv4 address family.

Note The 6VPE feature does not support per packet load sharing.

  For information on configuring this feature, see the How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN section in the *BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN* guide at:

  http://www.cisco.com/en/US/docs/ios/iproute/configuration/guide/irp_bgp_ebgp_ibgp.html#wp1054087

- VRF-aware BGP Dampening

  The 6VPE feature supports the same per-VRF BGP dampening mechanism as supported for IPv4 VPN, so that BGP Dampening can be separately controlled for each VRF.

  For information on configuring this feature, see the Configuring Route Dampening section in the Configuring Internal BGP Features chapter of the *Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4* guide at:

  http://www.cisco.com/en/US/docs/ios/12_4/ip_route/configuration/guide/1cfbgph.html#wp1002395

IPv6 Internet Access

Most VPN sites require access to the Internet. The following Internet access models are supported by IPv6 VPNs for enabling VPN access to the Internet.

- Model 1: Non-VRF Internet Access case.

  In some VPNs, one or more of the sites can obtain Internet access using an Internet gateway, such as a firewall, attached to a non-VRF interface to an Internet service provider (ISP). The ISP may or may not be the same organization as the Service Provider (SP) that is providing the VPN service. Traffic to or from the Internet gateway can be then routed according to the PE router's default forwarding table.

- Model 2: Some VPNs may obtain Internet access via an VRF interface.

  If a packet is received by a PE over a VRF interface and the packet's destination address does not match any route in the VRF, the packet can be matched against the PE's default forwarding table. If the packet matches the PE’s default forwarding table, the packet can be forwarded natively through

the backbone to the Internet instead of being forwarded by MPLS. In this model, the default forwarding table might have the full set of Internet routes, or it might have just a single default route leading to another router that does have the full set of Internet routes in its default forwarding table.

- Model 3: Using Static Routes in VRF that can be resolved using IPv6 Global Table.

  The Static Routes in VRFs can be resolved in the IPv6 Global Table in the same way they are currently supported for IPv4 VPN. Therefore, the network administrator can add in the IPv6 VRF, static routes, such as a default route, that points to an IPv6 Internet Gateway for outbound traffic from CE to Internet.

- Model 4: All Internet Routes in VRF.

  It is possible to obtain Internet access via a VRF interface by having the VRF include the Internet routes. This model involves redistributing the Internet Routes into the VRF.

VRF-aware Router Applications

The following features are supported on Cisco 10000 series router by the IPv6VPN over MPLS (6VPE) feature:

- VRF-aware Ping

  The VRF-aware Ping **ping vrf <VRF name> <IPv6-address>** command is supported:

- VRF-aware Traceroute

  The VRF-aware Traceroute **traceroute vrf <VRF name> < IPv6-address >** command is supported:

- VRF-aware Telnet

  The VRF-aware Telnet **telnet vrf <VRF name> < IPv6-address >** command is supported:

VRF-Lite

VRF-lite, also known as Multi-VRF CE, is an extension of IP routing with multiple routing instances on a CE Router. The VRF-lite feature does the following functions:

- Enables the creation of a Layer 3 VPN service by keeping separate IP routing and forwarding tables for each VPN customer.
- Uses input interfaces to distinguish routes for different VPNs.
- Forms virtual packet-forwarding tables by associating one or more interfaces with each VRF. An interface cannot belong to more than one VRF at any time.
- Supports overlapping unicast IP addresses across different VRFs.

VRF-lite is typically deployed along with Multiprotocol Label Switching (MPLS) VPN at the customer edge to support multiple customers on a single switch. The 6VPE feature supports the VRF-Lite feature in the same way as the feature is currently supported by IPv4 VPNs.

QoS features

The following features are supported on Cisco 10000 series router by the IPv6VPN over MPLS (6VPE) feature:

- Diff-Serv on Ingress PE

The 6VPE feature supports the same QoS mechanisms for IPv6 VPNs that is currently supported for IPv4 VPNs on the Ingress PE.

- Diff-Serv on Egress PE

  The 6VPE feature supports the same QoS mechanisms for IPv6 VPNs that is currently supported for IPv4 VPNs on the Egress PE.

- FRF.12

  The 6VPE feature supports FRF.12 fragmentation and interleaving for IPv6 traffic, in addition to IPv4 traffic, on PE to CE Frame Relay connections.

  For configuration tasks, see the FRF.12 Fragmentation section in the Fragmenting and Interleaving Real-Time and Nonreal-Time Packets chapter of the *Cisco 10000 Series Router Quality of Service Configuration Guide* at:

  http://www.cisco.com/en/US/docs/routers/10000/10008/configuration/guides/qos/10qlfi.html#wp1043021

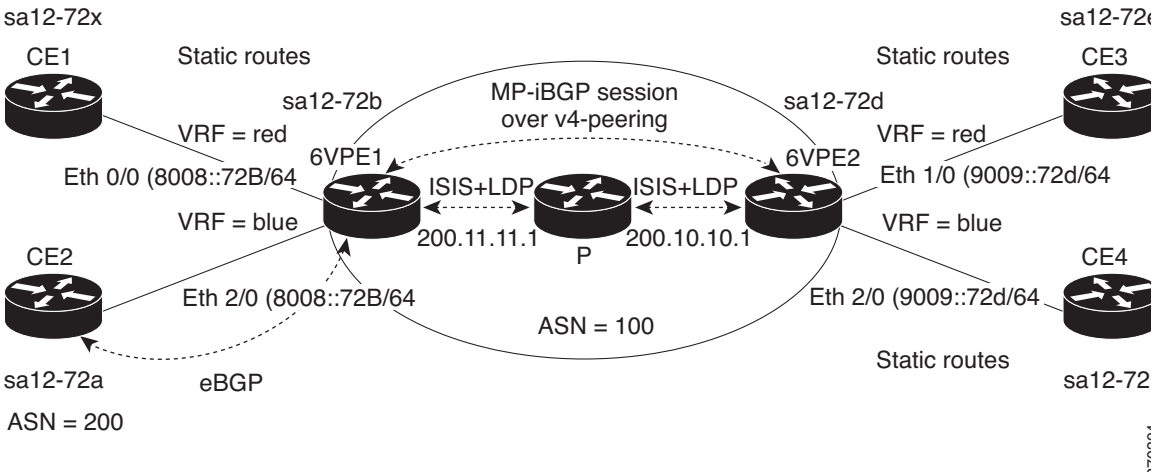
Note When an IPv6 packet arrives on an input interface configured for IPv6, either the packet has a Differentiated Services Code Point (DSCP) value set or an IPv6 QoS setup is done on the router to mark the DSCP value. This packet sent over a MPLS output interface will get the DSCP value mapped to the MPLS Experimental (EXP) bits. The mapping propagates the IPv6 QoS value to its MPLS equivalent.

Tip See the *Configuring a Basic MPLS VPN* document for setting up an IPv4 MPLS core network at: http://www.cisco.com/en/US/tech/tk436/tk428/technologies_configuration_example09186a00800a6c11.shtml

Complete 6VPE Configuration Example for Implementing IPv6 VPN over MPLS

The Figure 4-2 explains the Example 4-4 given below

Figure 4-2 IPv6 VPN over MPLS



Example 4-4 Configuring IPv6 VPN over MPLS

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname sa14-72b
!
logging snmp-authfail
logging queue-limit 100
!
clock timezone GMT 0
ip subnet-zero
ip cef
!
ipv6 unicast-routing
vrf definition blue
 rd 200:1
 address-family ipv6
 route-target export 200:1
 route-target import 200:1
 exit-address-family
!
vrf definition red
 rd 100:1
 address-family ipv6
 route-target export 100:1
 route-target import 100:1
 exit-address-family
!
ipv6 cef
mpls ldp logging neighbor-changes
mpls ldp router-id Loopback0
!
!
interface Loopback0
 ip address 200.11.11.1 255.255.255.255
 ipv6 address BEEF:11::1/64
 ipv6 nd prefix default 0 0 off-link no-autoconfig
 no ipv6 mfib fast
!
interface Ethernet0/0
 vrf forwarding red
 ip address 50.1.1.2 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 ipv6 address 4000::72B/64
 ipv6 address 8008::72B/64
 ipv6 nd prefix default infinite infinite
 no ipv6 mfib fast
!
interface Ethernet1/0
 ip address 40.1.1.2 255.255.255.0
 ip router isis
 no ip mroute-cache
 mpls ip
!
interface Ethernet2/0
 vrf forwarding blue
 ip address 90.1.1.2 255.255.255.0
 ipv6 address 8008::72B/64
 no ipv6 mfib fast
```

```
!
router isis
 net 49.0000.0000.0002.00
 redistribute connected metric 50
 passive-interface Loopback0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 200.10.10.1 remote-as 100
 neighbor 200.10.10.1 update-source Loopback0
 neighbor 8008::72a remote-as 200
 !
 address-family ipv4
 neighbor 200.10.10.1 activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 multicast
 no auto-summary
 exit-address-family
 !
 address-family vpnv6
 neighbor 200.10.10.1 activate
 neighbor 200.10.10.1 send-community extended
 exit-address-family
 !
 address-family ipv6 vrf red
 no synchronization
 redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf blue
 neighbor 8008::72a activate
 no synchronization
 redistribute connected
 exit-address-family
!
ip classless
no ip http server
!
end
```

Monitoring and maintaining IPv6 VPN over MPLS

For information on monitoring and maintaining IPv6 VPN over MPLS, see the Verifying and Troubleshooting IPv6 VPN section in the Implementing IPv6 VPN over MPLS (6VPE) chapter of the *Cisco IOS IPv6 Configuration Guide, Release 12.4T* guide at:

http://www.cisco.com/en/US/docs/ios/12_2t/ipv6/SA_vpnv6_ps6441_TSD_Products_Configuration_Guide_Chapter.html#wp1078529

Session Limit Per VRF

The session limit Per VRF feature enables you to limit the number of sessions that can be established for VPDN groups associated with a specific VPDN template. Previously, you associated all VPDN groups configured on the router with a single template. By using the session limit Per VRF feature, you can create, define, and name multiple VPDN templates, including a default VPDN template. You can

then associate a VPDN group with a specific VPDN template. By configuring a group session limit for a VPDN template, you can limit the maximum number of concurrent sessions allowed for all VPDN groups associated with the VPDN template.

If you configure a group session limit for the default VPDN template (the unnamed VPDN template), that session limit is the same for all VPDN groups not associated with a named VPDN template. If you configure a group session limit that is less than the number of current active sessions, no sessions are terminated and no new sessions can start. For example, if you configure a group session limit of 30 and 50 sessions are active, the router does not terminate any active sessions and it does not allow any new sessions to start.

If a VPDN group is associated with a VPDN template and the VPDN group has a session limit configured, the value of the VPDN group session limit takes precedence over the VPDN template session limit if the VPDN group value is less than the VPDN template value.

You can associate a VPDN group with only one VPDN template at a time. If you associate a VPDN group with a named VPDN template and then with a second VPDN template, the VPDN group is detached from the first VPDN template and associated with the second.

If you attempt to associate a VPDN group with a named VPDN template that you have not configured, the VPDN group uses the system defaults.

The **session-limit** global configuration command takes precedence over the **group session-limit** VPDN template configuration command. The **session-limit** command limits the number of VPDN sessions and the **group session-limit** command specifies the maximum concurrent sessions allowed across all VPDN groups associated with a particular VPDN template.

The Session Limit Per VRF feature is described in the following topics:

Application of VPDN Parameters to VPDN Groups

By default, the router applies VPDN parameters to a VPDN group in the following way:

- VPDN parameters configured for the individual VPDN group are always applied to that VPDN group.
- VPDN parameters configured in the VPDN template are applied for any setting not specified in the individual VPDN group configuration.
- System default settings for VPDN parameters are applied for any setting not configured in the individual VPDN group or VPDN template.

When you detach a VPDN group from a VPDN template by using the **no source vpdn-template** command, the router applies VPDN parameters to that VPDN group in the following way:

- VPDN parameters configured for the individual VPDN group are always applied to that VPDN group.
- System default settings for VPDN parameters are applied for any settings not configured in the individual VPDN group or VPDN template.

VPDN Template Configuration

Not all commands that are available for configuring a VPDN group can be used to configure a VPDN template. For a list of the commands available for VPDN template configuration, see the **vpdn-template** command reference page in the *Session Limit Per VRF*, Release 12.2(4)B feature module.

Feature History for Session Limit Per VRF

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was integrated into Cisco IOS Release 12.2(15)BX.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for Session Limit Per VRF

The session limit Per VRF feature has the following restrictions:

- Nesting of VPDN templates is not supported. You can associate a VPDN group with only one VPDN template at a time. If you associate a VPDN group with a named VPDN template and then with a second VPDN template, the VPDN group is detached from the first VPDN template and associated with the second template.
- If you attempt to associate a VPDN group with a named VPDN template that you have not configured, the VPDN group uses the system defaults.
- The **session-limit** global configuration command takes precedence over the **group session-limit** VPDN template configuration command.
- If the VPDN group value is less than the VPDN template value, the VPDN group session limit takes precedence over the VPDN template session limit.

Prerequisites for Session Limit Per VRF

The session limit Per VRF feature has the following requirements:

- You must have a VPDN enabled on the router and at least one VPDN group configured. The router must make an L2TP connection before VPDN configurations can be established.

Configuring Session Limit Per VRF

To configure the session limit Per VRF feature on the Cisco 10000 series router, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables virtual private dialup networking (VPDN) on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server if one is present.
Step 2	`Router(config)# vpdn session-limit` *sessions*	Limits the number of simultaneous VPN sessions that can be established on a router. The *sessions* option is the maximum number of simultaneous VPN sessions that you want to allow on a router. Valid values are 1 to 10,000.
Step 3	`Router(config)# vpdn-template` *template-name*	Configures a VPDN template and enters VPDN group configuration mode. The *template-name* option is the name of the VPDN template
Step 4	`Router(config-vpdn)# group session-limit` *number*	Specifies the maximum concurrent sessions allowed across all VPDN groups associated with the VPDN template you specified in step 3. The *number* option is a value from 1 to 32,767.
Step 5	Repeat steps 2 and 3 to configure additional named VPDN templates.	
Step 6	`Router(config-vpdn)# exit`	Exits VPDN group configuration mode.
Step 7	`Router(config)# vpdn-group` *tag*	Associates a VPDN group to a customer or VPDN profile. The *tag* option is the name of the VPDN group.

Step 8	`Router(config-vpdn)# accept-dialin`
	or
	`Router(config-vpdn)# request-dialout`
		Enables the router to accept dialin requests and enters VPDN accept-dialin group configuration mode.
		Enables the router to send L2TP dialout requests and enters VPDN request-dialout group configuration mode.

Step 9	`Router(config-vpdn-acc-in)# protocol` *protocol* or `Router(config-vpdn-req-out)# protocol` *protocol*	Specifies the tunneling protocol to be used.

Step 10	`Router(config-vpdn-acc-in)# exit`
	or
	`Router(config-vpdn-req-out)# exit`
		Exits VPDN accept-dialin or VPDN request-dialout group configuration mode.

Step 11	`Router(config-vpdn)# source vpdn-template` *template-name*	Configures the VPDN group to use the VPDN template settings for all unspecified parameters. The *template-name* option is the name of the VPDN template to be associated with a VPDN group.

	Command	Purpose
Step 12	`Router(config-vpdn)# session-limit session-number`	Limits the number of sessions allowed on the VPDN group. The *session-number* option is the maximum number of sessions allowed on the specified VPDN group. Valid values are from 0 to 32,767.
Step 13	Repeat steps 7 through 12 to configure session limiting on additional VPDN groups.	

Verifying a Session Limit Per VRF Configuration

To verify the configuration of the session limit Per VRF feature, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the current configuration of the router. Check the output of this command to confirm the configuration of a VPDN template group.
`Router# show vpdn session`	Displays the status of all active tunnels.

Configuration Examples for Session Limit Per VRF

Example 4-5 creates three VPDN groups named *group1*, *group2*, and *group3*. VPDN group1 and group2 are attached to the default VPDN template, which has a session limit of 10. VPDN group1 and group2 can have no more than a combined total of 10 concurrent sessions. For example, if group1 has three sessions, group2 can only have seven sessions.

In Example 4-5, using the **session-limit 5** command allows VPDN group1 to have no more than 5 sessions. Using the **session-limit 20** command allows VPDN group2 to have no more than 20 sessions. However, as previously indicated, the default VPDN template has a session limit of 10. Therefore, the combined number of sessions for VPDN group1 and group2 cannot exceed 10 sessions. If group1 has 5 sessions, group2 can only have 5 sessions. If group1 does not have any active sessions, group2 can have a maximum of 10 sessions, even though group2 is configured with the **session-limit 20** command.

In Example 4-5, VPDN group3 does not have a session limit configured. Using the **no source vpdn-template** command detaches group3 from the default VPDN template.

Example 4-5 Configuring Session Limit Per VRF

```
vpdn-template
 group session-limit 10
 exit

vpdn-group group2
 accept-dialin
  protocol any
  exit
 session-limit 20
 exit

vpdn-group group1
 accept-dialin
  protocol any
```

```
  exit
 session-limit 5

vpdn-group group3
 accept-dialin
  protocol any
  exit
 no source vpdn-template
```

Example 4-6 creates a default VPDN template and three VPDN groups named groupA, groupB, and groupC. As indicated in the default VPDN template configuration, the maximum combined number of sessions allowed for all VPDN groups associated with the default template is 10 sessions. The local name of the default VPDN template is local-name. Example 4-6 also creates an additional VPDN template named templateA, which has a session limit of 50. The combined number of sessions for all VPDN groups associated with templateA cannot exceed 50 concurrent sessions, regardless of the session limits set for the individual VPDN groups. VPDN groupA and groupB are attached to VPDN templateA and each group has an individual session limit of 30 sessions. Because groupA and groupB are attached to VPDN templateA, they use the hostname host1 as their local name.

In Example 4-6, the **source vpdn-template** command is not used to associate VPDN groupC with a specific VPDN template. Therefore, by default, VPDN groupC is attached to the default VPDN template, which has a group session limit of 10. VPDN groupC inherits the local name local-name from the default VPDN template.

Example 4-6 Configuring Session Limit Per VRF

```
hostname host1
vpdn-template
 group session-limit 10
 local name local-name
 exit

vpdn-template templateA
 group session-limit 50
 exit

vpdn-group groupA
 accept-dialin
  protocol any
  exit
 source vpdn-template templateA
 session-limit 30
 exit

vpdn-group groupB
 accept-dialin
  protocol any
  exit
 source vpdn-template templateA
 session-limit 30
 exit

vpdn-group groupC
 accept-dialin
  protocol any
```

Monitoring and Maintaining Session Limit Per VRF

To monitor and maintain the session limit Per VRF feature, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# show vpdn session [all [interface | tunnel | username] | packets | sequence | state | timers | window]`	Displays VPDN session information including interface, tunnel, username, packets, status, and window statistics. The options are: • **all**—All session information for active sessions • **all interface**—Interface associated to a specific session • **all tunnel**—Tunnel attribute filter • **all username**—Username filter • **packets**—Packet and byte count • **sequence**—Sequence numbers • **state**—State of each session • **timers**—Timer information • **window**—Window information
`Router# show vpdn`	Displays a summary of all active VPDN tunnels.
`Router# show vpdn group name`	Displays the session limit set and the number of active sessions and tunnels on the VPDN group you specify.
`Router# show vpdn history failure`	Displays information about VPDN user failures.

Half-Duplex VRF

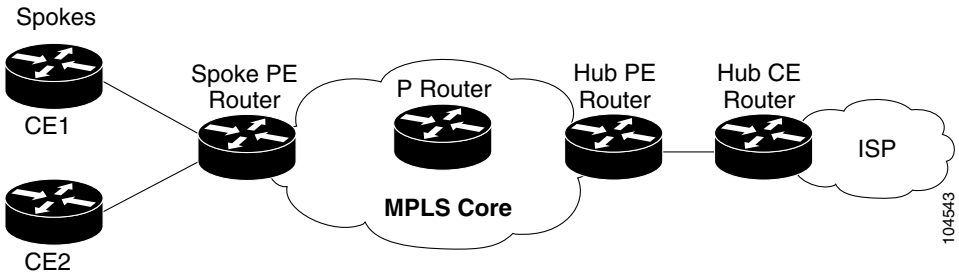
The Half-Duplex VRF (HDVRF) feature provides scalable hub and spoke connectivity for subscribers of a multiprotocol label switching-based virtual private network (MPLS VPN) service. These subscribers connect to the provider edge (PE) router of the wholesale service provider, and they use the same or different services (for example, the same or different VRFs). The HDVRF feature prevents local connectivity between subscribers at the spoke PE router and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE router must forward intersite traffic using the hub site. This ensures that the routing done at the spoke site is always access side interface to network side interface, or network side interface to access side interface, and never access side to access side.

In hub and spoke topologies in which multiple-spoke customer edge (CE) routers, also referred to as spokes, connect to the same PE router, the PE router locally switches the spokes without passing the traffic through the upstream Internet service provider (ISP). In releases earlier than Cisco IOS Release 12.2(16)BX2, when spokes connect to the same PE router, it was necessary to configure each spoke in a separate VRF to ensure that the traffic between the spokes always traverses the central link between the wholesale service provider and the ISP. However, this solution is manageable only if the number of spokes is relatively small. When a large number of spokes are connected to the same PE router, configuring a single VRF for each spoke can become quite complex and can greatly increase memory usage. This is true especially in large-scale wholesale service provider environments that support high-density remote access to Layer 3 VPNs.

The HDVRF feature addresses the limitations previously imposed on hub and spoke topologies by removing the requirement of one VRF per spoke and ensuring that subscriber traffic always traverses the central link between the wholesale service provider and the ISP, whether the subscriber traffic is being routed to a remote network by way of the upstream ISP or to another locally or remotely connected subscriber.

Figure 4-3 shows a sample hub and spoke topology for HDVRF.

Figure 4-3 Hub and Spoke Topology for Half-Duplex VRF



The Half-Duplex VRF feature is described in the following topics:

- Upstream and Downstream VRFs, page 4-21
- Reverse Path Forwarding Check Support, page 4-22
- Feature History for Half-Duplex VRF, page 4-22
- Restrictions for Half-Duplex VRF, page 4-22
- Prerequisites for Half-Duplex VRF, page 4-22
- Configuration Tasks for Half-Duplex VRF, page 4-23
- Configuration Examples for Half-Duplex VRF, page 4-25
- Monitoring and Maintaining Half-Duplex VRF, page 4-28

Upstream and Downstream VRFs

HDVRF uses two unidirectional VRFs, called upstream VRF and downstream VRF, to forward IP traffic between the spokes and the hub PE router.

The upstream VRF is used to forward the IP traffic from the spokes toward the MPLS VPN backbone. This VRF typically contains only a default route; but, depending on the configuration, it might also contain such information as summary routes and multiple default routes. The default route points to the interface on the hub PE router that connects to the upstream ISP. The Cisco 10000 series router dynamically learns about the default route from the routing updates that the hub PE router or home gateway sends. The upstream VRF also contains the virtual access interfaces that connect the spokes, but it contains no other local interfaces.

The downstream VRF is used to forward the traffic from the MPLS core back to the spokes. This VRF contains PPP peer routes for the spokes and per-user static routes imported from the authentication, authorization, and accounting (AAA) server. It also contains the routes imported from the hub PE router. These routes are the dynamically allocated virtual access interfaces of the subscribers associated with a particular service.

The Cisco 10000 series router redistributes routes from the downstream VRF into Multiprotocol Border Gateway Protocol (MP-BGP). The spoke PE router typically advertises a summary route across the MPLS core for the connected spokes. The upstream VRF configured on the hub PE router imports the advertised summary route.

Reverse Path Forwarding Check Support

Reverse Path Forwarding (RPF) check ensures that an IP packet entered the router using the correct inbound interface. The HDVRF feature supports unicast RPF check on the spoke-side interfaces. Because different VRFs are used for downstream and upstream forwarding, HDVRF extends the RPF mechanism to ensure that source address checks occur in the downstream VRF.

Feature History for Half-Duplex VRF

Cisco IOS Release	Description	Required PRE
12.2(16)BX2	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for Half-Duplex VRF

The Half-Duplex VRF feature has the following restrictions:

- In both the upstream and downstream VRFs, routing protocols are not supported on interfaces configured for half-duplex VRFs.
- Half-duplex VRFs apply only to virtual access interfaces (VAIs) and virtual template interfaces. Only IP unnumbered interfaces are supported.
- It is not supported with Routing with Bridged Encapsulation (RBE).

Prerequisites for Half-Duplex VRF

The Half-Duplex VRF feature has the following requirements:

- The spoke PE routers must be running Cisco IOS Release 12.2(16)BX2, Cisco IOS Release 12.3(7)XI1, or a later release.
- The performance routing engine (PRE), part number ESR-PRE2, must be installed in the router's chassis.

Configuration Tasks for Half-Duplex VRF

To configure the Half-Duplex VRF feature, perform the following configuration tasks:

- Configuring the Upstream and Downstream VRFs on the L2TP Access Concentrator and PE Router, page 4-23
- Associating VRFs, page 4-24
- Configuring RADIUS, page 4-25

Configuring the Upstream and Downstream VRFs on the L2TP Access Concentrator and PE Router

To configure the upstream and downstream VRFs on the PE router, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the VRF instance by assigning a VRF name.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.
Step 3	`Router(config-vrf)# route-target {import | export | both} route-target-ext-community`	Creates a list of import and export route target communities for the specified VRF. The **import** keyword is required to create an upstream VRF. The upstream VRF is used to import the default route from the hub PE router. The **export** keyword is required to create a downstream VRF. The downstream VRF is used to export the routes of all subscribers of a given service that the VRF serves.

Example 7 shows how to configure a downstream VRF named D.

Example 7 Configuring the Downstream VRF

```
Router(config)# ip vrf D
Router(config-vrf)# description Downstream VRF - to subscribers
Router(config-vrf)# rd 1:8
Router(config-vrf)# route-target export 1:100
```

Example 8 shows how to configure an upstream VRF named U.

Example 8 Configuring the Upstream VRF

```
Router(config)# ip vrf U
Router(config-vrf)# description Upstream VRF - to hub PE
Router(config-vrf)# rd 1:0
Router(config-vrf)# route-target import 1:0
```

Associating VRFs

After you define and configure the VRFs on the PE routers, associate each VRF with:

- An interface or subinterface, or
- A virtual template interface

The virtual template interface is used to create and configure a virtual access interface (VAI). For information about configuring a virtual template interface, see the "Configuring a Virtual Template Interface" section on page 3-17.

To associate a VRF, enter the following commands on the PE router beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip vrf forwarding vrf-name`	Associates an interface with the VRF you specify. *vrf-name* is the name of the VRF associated with the interface.
Step 2	`Router(config-if)# ip unnumbered type number`	Enables IP processing on an interface without assigning an explicit IP address to the interface. The *type* and *number* arguments are the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface. **Note** The Cisco 10000 series router supports only unnumbered interfaces for the Half-Duplex VRF feature.
Step 3	`Router(config-if)# exit`	Returns to global configuration mode.
Step 4	`Router(config)# interface virtual-template number`	Creates a virtual template interface and enters interface configuration mode.
Step 5	`Router(config-if)# ip vrf forwarding vrf-name1 [downstream vrf-name2]`	Associates a virtual template interface with the VRF you specify. The *vrf-name1* argument is the name of the VRF associated with the virtual template interface. The *vrf-name2* argument is the name of the downstream VRF into which the PPP peer route and all of the per-user routes from the AAA server are installed. If a AAA server is used, the AAA server provides the VRF membership; you do not need to configure the VRF members on the virtual templates.

Example 9 associates the VRF named vpn1 with the Virtual-Template1 interface and specifies the downstream VRF named D.

Example 9 Associating a Downstream VRF with a Virtual Template Interface

```
Router(config)# interface Virtual-Template1
Router(config-if)# ip vrf forwarding vpn1 downstream D
Router(config-if)# ip unnumbered Loopback1
Router(config-if)# no peer default ip address
Router(config-if)# ppp authentication chap vpn1
Router(config-if)# ppp authorization vpn1
Router(config-if)# ppp accounting vpn1
```

Configuring RADIUS

To configure the downstream VRF for an AAA server, enter the following Cisco attribute value:

```
cisco-avpair = "ip:vrf-id=vrf-name1 downstream vrf-name2"
```

where:

The *vrf-name1* argument is the name of the VRF associated with the subinterface or virtual template interface.

The *vrf-name2* argument is the name of the downstream VRF into which all of the subscriber routes from the AAA server are installed.

Note Instead of using the **lcp:interface-config** RADIUS attribute, we recommend that you use the **ip:vrf-id** RADIUS attribute when supported in Cisco IOS software. Unlike the **lcp:interface-config** attribute, which causes full virtual interfaces to be used, the **ip:vrf-id** attribute causes virtual subinterfaces to be used, which significantly improves scalability.

Example 10 shows how to configure a downstream VRF named D on a AAA server:

Example 10 Configuring the Downstream VRF on RADIUS

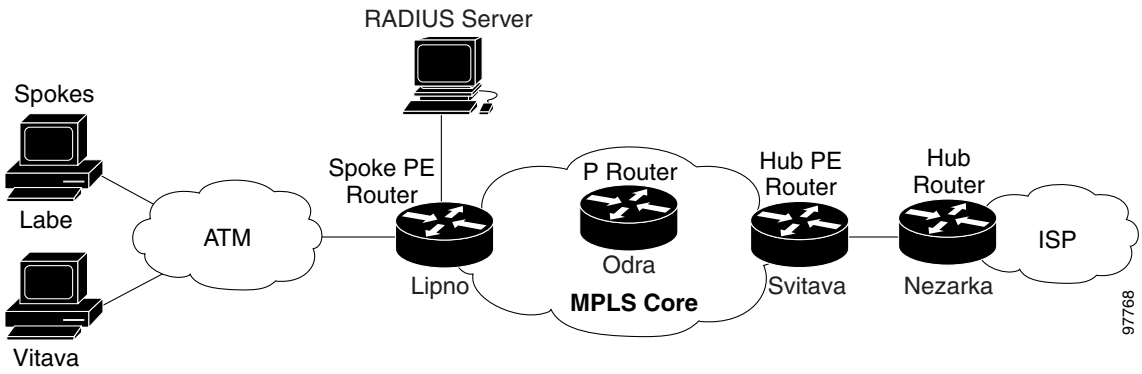
```
cisco-avpair = "ip:vrf-id=U downstream D"
```

Configuration Examples for Half-Duplex VRF

This section provides the following configuration examples. These examples use the hub and spoke topology shown in Figure 4-4.

- Hub and Spoke Sample Configuration with Half-Duplex VRFs, page 4-26
- RADIUS Sample Configuration, page 4-27

Figure 4-4 Sample Topology for Half-Duplex Configuration



Hub and Spoke Sample Configuration with Half-Duplex VRFs

Example 4-11 shows how to connect two PPPoE clients to a single VRF pair on the spoke PE router named *Lipno*. Although both PPPoE clients are configured in the same VRF, all communication occurs using the hub PE router. Half-duplex VRFs are configured on the spoke PE. The client configuration is downloaded to the spoke PE from the RADIUS server.

Example 4-11 Configuring the Spoke PE Router

```
aaa new-model
!
aaa group server radius R
 server 22.0.20.26 auth-port 1812 acct-port 1813
!
aaa authentication ppp default group radius
aaa authorization network default group radius
!
ip vrf D
 description Downstream VRF - to spokes
 rd 1:8
 route-target export 1:100
!
ip vrf U
 description Upstream VRF - to hub
 rd 1:0
 route-target import 1:0
!
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback0
ip address 100.0.0.8 255.255.255.255
!
interface Loopback2
ip unnumbered Loopback2
 ip vrf forwarding U
 ip address 2.0.0.8 255.255.255.255
!
interface ATM2/0
 pvc 3/100
  protocol pppoe
 !
 pvc 3/101
  protocol pppoe
 !
interface Virtual-Template1
 no ip address
 ppp authentication chap
!
router bgp 1
 no synchronization
 neighbor 100.0.0.34 remote-as 1
 neighbor 100.0.0.34 update-source Loopback0
 no auto-summary
 !
address-family vpnv4
  neighbor 100.0.0.34 activate
  neighbor 100.0.0.34 send-community extended
```

```
  no auto-summary
  exit-address-family
 !
 address-family ipv4 vrf U
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family ipv4 vrf D
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
ip local pool U-pool 2.8.1.1 2.8.1.100
!
radius-server host 22.0.20.26 auth-port 1812 acct-port 1813
radius-server key cisco
```

RADIUS Sample Configuration

Example 4-12 shows how to configure the RADIUS server for HDVRF support. In this example, the spokes inherit the default configuration. Static routes per spoke are defined to demonstrate that HDVRF supports per-user static routes. The functionality of the HDVRF feature does not require that you define static routes per spoke. This configuration was tested on FreeRADIUS 0.8.1.

Example 4-12 Configuring RADIUS for Half-Duplex VRFs

```
DEFAULT Service-Type == Framed-User
        Framed-Protocol = PPP,
        cisco-avpair = "ip:vrf-id=U downstream D",
        cisco-avpair = "ip:ip-unnumbered=Loopback 2",
        cisco-avpair = "ip:addr-pool=U-pool",
        Fall-Through = Yes

labe    Auth-Type := Local, User-Password == "labe"
        cisco-avpair = "ip:route=2.0.0.5 255.255.255.255"

vltava  Auth-Type := Local, User-Password == "vltava"
        cisco-avpair = "ip:route=2.0.0.2 255.255.255.255"
```

Note Instead of using the **lcp:interface-config** RADIUS attribute, we recommend that you use the **ip:vrf-id** RADIUS attribute when supported in Cisco IOS software. Unlike the **lcp:interface-config** attribute, which causes full virtual interfaces to be used, the **ip:vrf-id** attribute causes virtual subinterfaces to be used, which significantly improves scalability.

Monitoring and Maintaining Half-Duplex VRF

To monitor and maintain upstream and downstream VRFs, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show cef interface virtual-interface number internal`	Displays internal information about the virtual access interface (VAI) you specify, including the downstream VRF associated with the VAI.
`Router# show ip interface virtual-interface number`	Displays information about the VAI you specify, including the downstream VRF associated with the VAI.
`Router# show ip route vrf vrf-name`	Displays the IP routing table for the VRF you specify. Use this command to display information about the per-user static routes installed in the downstream VRF.
`Router# show ip vrf`	Displays information about all of the VRFs configured on the router, including the downstream VRF for each associated VAI.
`Router# show ip vrf detail vrf-name`	Displays detailed information about the VRF you specify, including all of the VAIs associated with the VRF. If you do not specify a value for *vrf-name*, detailed information about all of the VRFs configured on the router appears, including all of the VAIs associated with each VRF.
`Router# show running-config interface type number`	Displays information about the virtual access interface you specify, including information about the upstream and downstream VRFs.

Example 4-13 shows how to display information about the interface named virtual-access 3.

Example 4-13 show running-config interface—virtual-access 3

```
Lipno# show running-config interface virtual-access 3

Building configuration...

Current configuration : 92 bytes
!
interface Virtual-Access3
 ip vrf forwarding U downstream D
 ip unnumbered Loopback2
end
```

Example 4-14 shows how to display information about the interface named virtual-access 4.

Example 4-14 show running-config interface—virtual-access 4

```
Lipno# show running-config interface virtual-access 4

Building configuration...

Current configuration : 92 bytes
!
interface Virtual-Access4
 ip vrf forwarding U downstream D
 ip unnumbered Loopback2
end
```

Example 4-15 shows how to display the routing table for the downstream VRF named D.

Example 4-15 show ip route vrf—Downstream

```
Lipno# show ip route vrf D

Routing Table: D
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
U       2.0.0.2/32 [1/0] via 2.8.1.1
S       2.0.0.0/8 is directly connected, Null0
U       2.0.0.5/32 [1/0] via 2.8.1.2
C       2.8.1.2/32 is directly connected, Virtual-Access4
C       2.8.1.1/32 is directly connected, Virtual-Access3
```

Example 4-16 shows how to display the routing table for the upstream VRF named U.

Example 4-16 show ip route vrf—Upstream

```
Lipno# show ip route vrf U


Routing Table: U
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 100.0.0.20 to network 0.0.0.0

     2.0.0.0/32 is subnetted, 1 subnets
C       2.0.0.8 is directly connected, Loopback2
B*   0.0.0.0/0 [200/0] via 100.0.0.20, 1w5d
```

CHAPTER 5

Configuring the Layer 2 Tunnel Protocol Access Concentrator and Network Server

The Cisco 10000 series router supports the Layer 2 Tunnel Protocol (L2TP) to allow users and telecommuters to connect to their corporate intranets or extranets. The Cisco 10000 series router supports the Layer 2 access concentrator (LAC) and Managed L2TP network server features. These features enable the Cisco 10000 series router to act as either a LAC or an LNS device.

Acting as the LAC, the Cisco 10000 router uses L2TP tunnels to forward packets to the LNS. As the LNS, the Cisco 10000 series router terminates and routes subscriber sessions into the appropriate virtual routing and forwarding (VRF) instance.

This chapter describes the following features:

IP Reassembly

The Cisco 10000 series router supports the IP Reassembly feature on the fastpath. This feature reassembles fragments of IP and L2TP encapsulated packets.

The IP Reassembly feature on the fastpath reassembles IP packets that have two IPv4 non-overlapping no-option fragments and drops two fragment overlapping fragments. The Route Processor (RP) handles packets with options, non-IPv4 packets, and packets with three or more fragments. If input security ACLs are configured, IP Reassembly processes the ACLs on the fragments and also on the reassembled packet.

Intermediate routers fragments an IP datagram if the outgoing maximum transmission unit (MTU) is lower than the packet size. The receiving host is responsible for reassembling the datagram from the fragments. When configured as a LAC, LNS, or tunnel switch, the Cisco 10000 series router is the receiving host for the tunneled packets. If one of the intermediate routers fragments L2TP encapsulated packets in transit through the tunnel, the IP Reassembly feature reassembles the packets.

Feature History for IP Reassembly

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Layer 2 Access Concentrator

The Cisco 10000 series router supports the Layer 2 access concentrator (LAC) feature. When configured as the LAC, the Cisco 10000 series router functions as the service provider's network access server. Remote subscribers use a local or point-to-point connection to initiate a PPPoA or PPPoE session to the LAC. The LAC terminates the physical connection and forwards the PPP session to the provider's Layer 2 Tunnel Protocol network server (LNS).

The LAC connects to the LNS using a local area network or a wide area network such as public or private ATM. The LAC directs subscriber sessions into Layer 2 Tunnel Protocol (L2TP) tunnels based on the domain of each session. The LAC acts as one side of an L2TP tunnel endpoint and is a peer to the LNS on the other side of the tunnel. The LAC forwards packets to and from the LNS and a remote system.

Acting as the LNS, you can configure the Cisco 10000 series router to terminate the PPP sessions and route the client IP packets onto the ISP or corporate network toward their final destination (Figure 5-1). You can also configure the LNS to place the sessions in VRFs before routing the packets, as shown in Figure 5-2.

The LAC feature is described in the following topics:

Figure 5-1 Terminating and Forwarding Sessions from the LAC

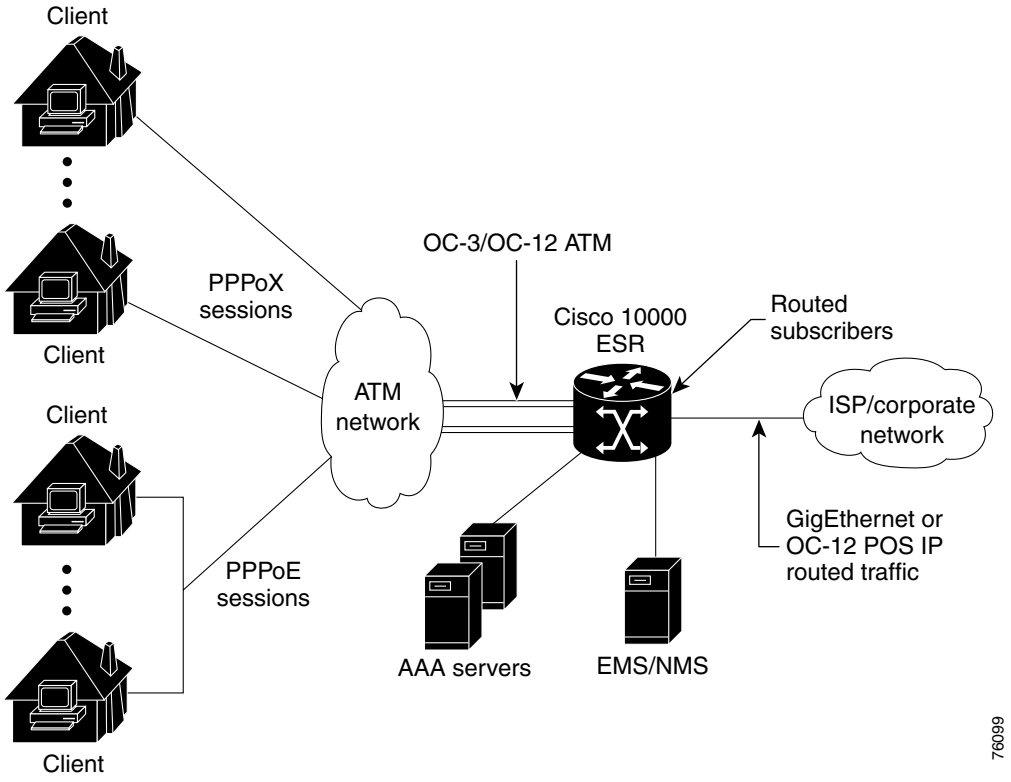
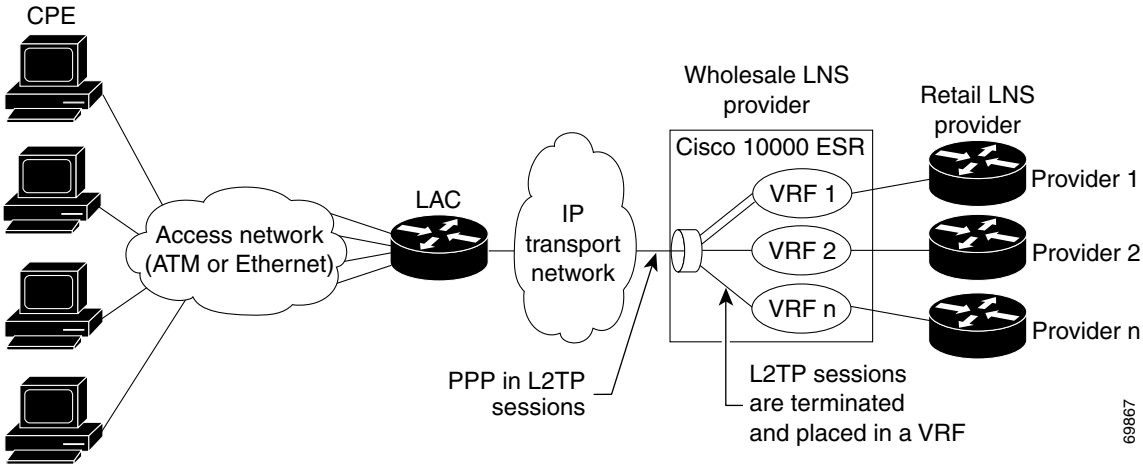


Figure 5-2 Placing Sessions from the LAC in VRFs



Tunnel Sharing

The tunnel sharing feature enables sessions that are authorized with different domains to share the same tunnel. Tunnel sharing reduces the number of tunnels required from the LAC. When used with the L2TP multihop feature, tunnel sharing also reduces the number of tunnels to an LNS. While improving tunnel management, tunnel sharing helps to reduce the number of tunnel establishment messages that are sent after interface dropouts, reducing dropout recovery time.

Note The session per tunnel limiting feature, when configured, limits the number of PPP sessions from multiple domain names that can be forwarded in a single tunnel.

The **domain** *domain-name* command in request-dialin or virtual private dial network (VPDN) group configuration mode requests that the LAC tunnel PPP sessions from a specific *domain-name*. Applying multiple instances of this command in a VPDN group or subgroup enables the LAC to forward PPP sessions from any of the specified domains in the same tunnel.

Tunnel Service Authorization

The tunnel service authorization feature allows the service provider to limit the number of destinations a subscriber can choose and to charge a fee for each destination allowed. The LAC can conduct static or dynamic tunnel service authorization.

A static domain name on an ATM PVC port overrides the domain name that the client session supplies. Static tunnel service authorization does not support switched virtual circuits (SVCs).

If a static domain is not configured, the LAC conducts dynamic tunnel service authorization. During dynamic tunnel service authorization, the LAC performs the following steps:

1. Domain Preauthorization—Checks the client-supplied domain name (in the PPP username) against an authorized list configured on the RADIUS server for each PVC.

   If the domain name is on the authorized list, the LAC proceeds to tunnel service authorization.

   If the domain name is not on the authorized list, the LAC attempts PPP authentication and authorization for local termination. The **vpdn authorize domain** command configures the domain preauthorization feature.

2. Tunnel Service Authorization—Checks the client-supplied domain name against a list of domains provided in the user profile on the RADIUS server to determine the domains accessible to the user. Enables tunnel service authorization and establishes an L2TP tunnel.

The following sections discuss tunnel selection as it relates to tunnel service authorization.

Tunnel Selection

When configured as the LAC, the Cisco 10000 series router selects a tunnel for an incoming PPP session using the following features:

- Static tunnel selection
- Per user tunnel selection
- Dynamic tunnel selection

Static Tunnel Selection

The static tunnel selection feature specifies a domain name for a PVC on an ATM interface. The LAC uses the specified domain name to select a tunnel for all PPP sessions originating from the PVC. This feature ignores the domains subscribers indicate in their usernames and forces the subscribers to a specific destination.

The **vpn service** *domain-name* command in ATM VC configuration mode configures the *domain-name* on the specified PVC. The **vpn service** *domain-name* command in ATM VC class configuration mode configures the *domain-name* on all virtual circuits in the VC class.

Per User Tunnel Selection

The per user tunnel selection feature specifies that the LAC use the entire structured PPP username to select a tunnel for forwarding an incoming session. Instead of sending the domain name, the LAC sends the entire structured PPP username to the authentication, authorization, and accounting (AAA) server. The AAA server provides the VPDN tunnel attributes for the user, indicating which tunnel the LAC can use to forward the session.

The **authen-before-forward** command in VPDN group configuration mode configures the per user tunnel selection feature.

Note When tunneling from a LAC to an LNS using L2TP, when you use the **authen-before-forward** command to configure the LAC to authenticate the user to RADIUS before negotiating a tunnel with the LNS, the user is authenticated and the LAC uses RADIUS information to determine if it should terminate a PPPoX session as PPP terminated aggregation (PTA) or forward the session to the LNS.

Dynamic Tunnel Selection

The dynamic tunnel selection feature enables the LAC to use the client-supplied domain in the PPP username to select a tunnel for forwarding an incoming session. You must configure a VPDN group on the LAC for each possible domain that a user might indicate.

Note You can restrict a user from certain domains by using domain preauthorization and tunnel service authorization. For more information, see the "Tunnel Service Authorization" section on page 5-4.

Sessions per Tunnel Limiting

The sessions per tunnel limiting feature specifies the maximum number of sessions initiated within an L2TP tunnel. The **initiate-to ip** command in VPDN group configuration mode configures the session per tunnel limiting feature. The command syntax is:

```
initiate-to ip ipaddress [limit limit-number] [priority priority-number]
```

Because the sessions per tunnel limiting feature enables you to specify the maximum number of VPDN sessions terminating at any L2TP network server (LNS), you can keep corporate router utilization at a more predictable level.

Session Load Balancing

The session load balancing feature enables the LAC to direct sessions across multiple LNS devices. The LAC retrieves L2TP tunnel (VPDN) information from local configuration or a RADIUS server. Both configuration methods support load balancing, but using RADIUS is more scalable than the local method. When you enable the session load balancing feature using RADIUS, the server sends L2TP tunnel information using multiple Tunnel-Server-Endpoint attributes in one tagged attribute group.

Multiple instances of the **initiate-to ip** command in VPDN group configuration mode configures the session load balancing feature locally. For information on the command syntax, see the "Sessions per Tunnel Limiting" section on page 5-5.

When you enable the session load balancing feature, the LAC uses a priority or round-robin load balancing algorithm to forward PPP sessions destined to the same domain among multiple tunnels.

Note Load balancing occurs with respect to the load a particular LAC generates. The LAC is not aware of the true load on a set of LNS devices. The true load on the LNS devices is an aggregation of all LAC devices using the LNS devices.

Session Load Failover

The session load failover feature works with the session load balancing feature to enable the LAC to direct sessions across multiple LNS devices. If the primary set of LNS devices fails, the session load failover feature enables the LAC to direct sessions to a set of failover LNS devices. The LAC uses the failover LNS devices only if *all* of the primary set of devices are unavailable. Failover occurs if the LAC:

- Sends an excessive number of Start-Control-Connection-Requests (SCCRQs) (no response from peer)
- Receives a Stop Control Channel (StopCNN) message from its peer during tunnel establishment
- Receives a Call-Disconnect-Notify (CDN) message during session establishment
- Receives vendor-specific attributes (VSAs) or standard RADIUS AV pairs indicating failover

The RADIUS Tunnel-Preference attribute is used to form load failover groups. When the values of the Tunnel-Preference attributes for different tagged attribute groups are the same, the Tunnel-Server-Endpoint for each of those attribute groups has the same failover priority.

Feature History for LAC

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for LAC

When configured as a LAC device, the Cisco 10000 series router has the following restrictions:

- The L2TP LAC per session features do not support PPP quality of service (QoS) and security access control lists (ACLs).
- The Cisco 10000 series router does not support the configuration of L2TP tunnels over the management Fast Ethernet interface. Do not set up L2TP tunnels over this interface.

Required Configuration Tasks for LAC

To configure the Cisco 10000 series router to act as a LAC, perform the following required configuration task:

- Enabling the LAC to Look for Tunnel Definitions, page 5-7

Enabling the LAC to Look for Tunnel Definitions

To enable the LAC to look for tunnel definitions, you must enable the VPDN feature on the LAC. To enable VPDN, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 1	`Router# config terminal`	Enters global configuration mode.
Step 1	`Router(config)# vpdn enable`	Enables VPDN networking on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server (home gateway), if one is present.
Step 2	`Router(config)# vpdn-group group-name`	Defines a local group name for which you can assign other VPDN variables. Enters VPDN group configuration mode.
Step 3	`Router(config-vpdn)# request-dialin`	Enables the LAC to request L2TP tunnels to the Cisco 10000 router and enters VPDN request-dialin group mode.
Step 4	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol.
Step 5	`Router(config-vpdn-req-in)# exit`	Returns to VPDN group configuration mode.
Step 6	`Router(config-vpdn)# initiate-to ip ip-address [priority priority-number]`	Specifies the LNS IP address and optionally the priority of the IP address (1 is the highest).

Optional Configuration Tasks for LAC

To configure the Cisco 10000 series router as a LAC, perform any of the following optional tasks:

- Enabling Sessions with Different Domains to Share the Same Tunnel, page 5-8
- Enabling the LAC to Conduct Tunnel Service Authorization, page 5-8
- Configuring Sessions Per Tunnel Limiting on the LAC, page 5-12

Enabling Sessions with Different Domains to Share the Same Tunnel

To enable sessions authorized with different domains to share the same tunnel, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# vpdn-group group-name`	Defines a local group name for which you can assign other VPDN variables. Enters VPDN group configuration mode.
Step 4	`Router(config-vpdn)# request-dialin`	Enables the LAC to request L2TP tunnels to the Cisco 10000 series router and enters VPDN request-dialin group mode.
Step 5	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol.
Step 6	`Router(config-vpdn-req-in)# domain domain-name`	Requests that PPP calls from the specified domain be tunneled. **Note** For multiple domains over the same tunnel, repeat this step to list all of the domains you want that tunnel to support. To configure the same domain over multiple tunnels, you must configure load balancing and sharing between the tunnels by using the **loadsharing ip** *ip-address* [**limit** *session-limit*] command in VPDN group configuration mode.
Step 7	`Router(config-vpdn-req-in)# exit`	Returns to VPDN group configuration mode.
Step 8	`Router(config-vpdn)# initiate-to ip ip-address [priority priority-number]`	Specifies the LNS IP address and optionally the priority of the IP address (1 is the highest).

Verifying Tunnel Sharing Configuration on the LAC

To verify tunnel sharing configuration on the LAC, enter the following command in privileged EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the running configuration and allows you to check that you successfully enabled the tunnel sharing feature.

Enabling the LAC to Conduct Tunnel Service Authorization

To enable the LAC to conduct static or dynamic tunnel service authorization, perform the following tasks:

- Configuring a Static Domain Name on a Permanent Virtual Circuit, page 5-8 or Configuring a Static Domain Name on a Virtual Circuit Class, page 5-10
- Enabling Domain Preauthorization, page 5-11
- Configuring the LAC to Communicate with the RADIUS Server, page 5-11

Configuring a Static Domain Name on a Permanent Virtual Circuit

To configure a static domain name on a permanent virtual circuit (PVC), enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# interface atm 0/0/0[.subinterface-number] {multipoint | point-to-point | tag-switching}`	Specifies the ATM interface and optional subinterface.
Step 4	`Router(config-subif)# atm pppatm passive`	Places the sessions on the subinterface in passive (listening) mode.
Step 5	`Router(config-subif)# no ip directed-broadcast`	Disables forwarding of directed broadcasts.
Step 6	`Router(config-subif)# pvc [name] vpi/vci`	Configures a PVC on the ATM interface or subinterface.
Step 7	`Router(config-if-atm-vc)# encapsulation aal5mux ppp Virtual-Template number`	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM PVC, SVC, VC class, or VC bundle. **mux ppp** is for a MUX-type VC running IETF-compliant PPP over ATM. You must use the **Virtual-Template** *number* argument to identify the virtual template. The **mux ppp** keyword applies to ATM PVCs only.
Step 8	`Router(config-if-atm-vc)# vpn service domain-name`	Configures the static domain name on the PVC.

Example 5-1 shows the static domain names *net1.com* and *net2.com* assigned to PVCs on an ATM interface. All PPP sessions originating from PVC 30/33 are sent to the *net1.com* L2TP tunnel. All PPP sessions originating from PVC 30/34 are sent to the *net2.com* tunnel.

Example 5-1 Configuring a Static Domain Name on a Permanent Virtual Circuit

```
!
interface ATM 0/0/0.33 multipoint
    atm pppatm passive
    pvc 30/33
        encapsulation aa15ciscoppp Virtual-Template1
        vpn service net1.com
    !
    pvc 30/34
        encapsulation aa15ciscoppp Virtual-Template1
        vpn service net2.com
    !
```

Configuring a Static Domain Name on a Virtual Circuit Class

To configure a static domain name on a VC class, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# vc-class atm` *vc-class-name*	Creates and names a map class.
Step 4	`Router(config-vc-class)# encapsulation aal5mux ppp Virtual-Template` *number*	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM PVC, SVC, VC class, or VC bundle. **mux ppp** is for a MUX-type VC running IETF-compliant PPP over ATM. You must use the **Virtual-Template** *number* argument to identify the virtual template. The **mux ppp** keyword applies to ATM PVCs only.
Step 5	`Router(config-vc-class)# vpn service` *domain-name*	Configures the static domain name on the VC class.
Step 6	`Router(config-vc-class)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# interface atm 0/0/0`[*.subinterface-number*] {**multipoint** | **point-to-point** | **tag-switching**}	Specifies the ATM interface and optional subinterface.
Step 8	`Router(config-subif)# atm pppatm passive`	Places the sessions on the subinterface in passive (listening) mode.
Step 9	`Router(config-subif)# class-int` *vc-class-name*	Applies the VC class to all VCs on the ATM interface or subinterface.

In Example 5-2, the static domain name *net.com* is assigned to a VC class. The VC class is then assigned to the VCs on an ATM subinterface.

Example 5-2 Configuring a Static Domain Name on a VC Class

```
!
vc-class ATM MyClass
    encapsulation aal5ciscoppp Virtual-Template1
    vpn service net.com
    !
interface ATM 0/0/0.99 multipoint
    atm pppatm passive
    class-int MyClass
    no ip directed-broadcast
    pvc 20/40
    pvc 30/33
    !
```

Verifying the Static Domain Name

To verify that you successfully configured the static domain name, enter the **show running-config** command in privileged EXEC mode.

Enabling Domain Preauthorization

To enable the LAC to perform domain authorization before tunneling, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# vpdn authorize domain`	Enables domain preauthorization.

Example 5-3 Enabling Domain Preauthorization

```
!
aaa new-model
aaa authorization network default local group radius
!
vpdn authorize domain
!
radius-server host 10.16.9.9 auth-port 1645 acct-port 1646
radius-server attribute nas-port format d
radius-server key MyKey
radius-server vsa send authentication
!
```

Verifying Domain Preauthorization

To verify that you successfully enabled domain preauthorization, enter the following commands:

Command	Purpose
`Router# show running-config`	Verifies that you successfully configured the maximum number of sessions per tunnel.
`Router# show vpdn tunnel`	Verifies active L2TP tunnel information in a VPDN environment.
`Router# show vpdn session`	Verifies active L2TP sessions in a VPDN environment.

Configuring the LAC to Communicate with the RADIUS Server

To enable the LAC to communicate properly with the RADIUS server for tunnel service authorization, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# radius-server host {`*hostname* `|` *ip-address*`} [auth-port` *port-number*`] [acct-port` *port-number*`]`	Specifies the RADIUS server host.
Step 4	`Router(config)# radius-server retransmit` *retries*	Specifies the number of times the Cisco IOS software searches the list of RADIUS server hosts before giving up. The default number of retries is 3 attempts.

	Command	Purpose
Step 5	`Router(config)# radius-server attribute 44 include-in-access-req vrf vrf-name`	Sends RADIUS attribute 44 (Accounting Session ID) in access request packets before user authentication (including requests for preauthentication).
Step 6	`Router(config)# radius-server domain-stripping vrf vrf-name`	(Optional) Enables VRF-aware domain-stripping. The **vrf** *vrf-name* argument specifies the per VRF configuration.
Step 7	`Router(config)# radius-server attribute list list-name`	Defines the list name given to the set of attributes defined using the **attribute** command.
Step 8	`Router(config)# radius-server key string`	Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 9	`Router(config)# radius-server vsa send authentication`	Configures the LAC to recognize and use vendor-specific attributes.

Example 5-4 Configuring Communication with the RADIUS Server

```
!
aaa new-model
aaa authorization network default local group radius
!
radius-server host 10.16.9.9 auth-port 1645 acct-port 1646
radius-server attribute 44 include-in-access-req vrf vrf1
radius-server key MyKey
radius-server vsa send authentication
```

Verifying Communication with the RADIUS Server

To verify that you successfully configured the LAC to communicate properly with the RADIUS server for tunnel service authorization, enter the **show running-config** command in privileged EXEC mode.

Configuring Sessions Per Tunnel Limiting on the LAC

To limit the number of sessions per tunnel without using a RADIUS server, enter the following commands.

Note You can configure the LAC or the RADIUS server to limit the number of sessions per tunnel. For information on using the RADIUS server for sessions per tunnel limiting, see the "Configuring Sessions Per Tunnel Limiting in the RADIUS Service Profile" section on page 5-16.

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# vpdn-group group-name`	Defines a local group name for which you can assign other VPDN variables. Enters VPDN group configuration mode.
Step 4	`Router(config-vpdn)# request-dialin`	Enables the LAC to request L2TP tunnels to the LNS and enters VPDN request-dialin group mode.
Step 5	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol.

	Command	Purpose
Step 6	`Router(config-vpdn-req-in)# domain domain-name`	Initiates a tunnel based on the client-supplied domain name.
Step 7	`Router(config-vpdn-req-in)# exit`	Returns to VPDN group mode.
Step 8	`Router(config-vpdn)# initiate-to ip ip-address limit limit-number [priority priority-number]`	Specifies the LNS IP address, the maximum number of sessions per tunnel, and optionally the priority of the IP address (1 is the highest).

Verifying Sessions Per Tunnel Limiting on the LAC

To verify sessions per tunnel limiting on the LAC, enter the following commands:

Command	Purpose
`Router# show running-config`	Verifies that you successfully configured the maximum number of sessions per tunnel.
`Router# show vpdn tunnel`	Verifies that the number of displayed sessions does not exceed your configured limit.

Example 5-5 Verifying Sessions Per Tunnel Limiting on the LAC

```
Router> enable
Router# show vpdn tunnel

L2TP Tunnel Information (Total tunnels 50 sessions 2000)

LocIDRemIDRemote NameStateRemote AddressPortSessions
412347811LNS1est10.16.1.1170140
200222323LNS1est10.16.1.1170140
412347811LNS2est10.16.2.2170140
597653477LNS2est10.16.3.3170140
!
!
```

RADIUS Server Optional Configuration Tasks for LAC

To configure the optional RADIUS server for the LAC, perform any of the following optional tasks:

- Enabling Tunnel Sharing for RADIUS Services, page 5-13
- Enabling the RADIUS Server to Conduct Tunnel Service Authorization, page 5-14
- Configuring Sessions Per Tunnel Limiting in the RADIUS Service Profile, page 5-16

Enabling Tunnel Sharing for RADIUS Services

To configure tunnel sharing in the RADIUS service profile, enter the following Cisco-AV pair attributes in the profile:

- vpdn-group
- tunnel-share

VPDN Group

The vpdn-group attribute specifies the group to which the service belongs. All services with matching group names are considered members of the same VPDN group. This attribute has the following syntax:

```
Cisco-AVpair="vpdn:vpdn-group=group-name"
```

group-name is the group to which the service belongs.

Example 5-6 VPDN Group—RADIUS Freeware Format

```
Cisco-AVpair="vpdn:vpdn-group=group1"
```

Tunnel Share

The tunnel-share attribute indicates that the tunnel sharing feature is enabled for the service.

Example 5-7 Tunnel Share—RADIUS Freeware Format

```
Cisco-AVpair="vpdn:tunnel-share=yes"
```

Verifying the Tunnel Sharing Configuration in the RADIUS Service Profile

To verify the RADIUS service profile, see the user documentation for your RADIUS server.

Enabling the RADIUS Server to Conduct Tunnel Service Authorization

To enable the RADIUS server to conduct dynamic tunnel service authorization, perform the following tasks:

- Configuring the RADIUS User Profile for Domain Preauthorization, page 5-14
- Configuring the RADIUS Service Profile for Tunnel Service Authorization, page 5-15

Configuring the RADIUS User Profile for Domain Preauthorization

To enable domain preauthorization, enter the following configuration parameters in the user profile on the RADIUS server:

RADIUS Entry	Purpose
nas-port:*ip-address*:*slot*/*subslot*/*port*/*vpi*.*vci*	Configures the NAS port username for domain preauthorization. The *ip-address* argument is the management IP address of the network service provider (NSP). The *slot*/*subslot*/*port* argument specifies the ATM interface. The *vpi*.*vci* arguments are the VPI and VCI values for the PVC.
`Password = "cisco"`	Sets the fixed password.
`User-Service-Type = Outbound-User`	Configures the service-type as outbound.
Cisco-AVpair = "vpdn:vpn-domain-list=domain1, domain2,..."	Specifies the domains accessible to the user.

Example 5-8 Configuring the RADIUS User Profile for Domain Preauthorization

```
user = nas-port:10.16.9.9:0/0/0/30.33{
    profile_id = 826
    profile_cycle = 1
    radius=Cisco {
    check_items = {
    2=cisco
    }
    reply_attributes= {
    9, 1="vpdn:vpd-domain-list=net1.com,net2.com"
```

Verifying the RADIUS User Profile for Domain Preauthorization

To verify the RADIUS user profile, see your RADIUS server user documentation.

Configuring the RADIUS Service Profile for Tunnel Service Authorization

To enable tunnel service authorization, enter the following configuration parameters in the service profile on the RADIUS server:

RADIUS Entry	Purpose
`domain` **`Password "cisco"`**	Sets the fixed password.
`User-Service-Type = Outbound-User`	Configures the service-type as outbound.
`Cisco-AVpair = "vpdn:tunnel-id=`*`name`***`"`**	Specifies the name of the tunnel that must match the LNS's VPDN terminate-from hostname.
`Cisco-AVpair = "vpdn:12tp-tunnel-password=`*`secret`***`"`**	Specifies the secret (password) for L2TP tunnel authentication.
`Cisco-AVpair = "vpdn:tunnel-type=12tp"`	Specifies Layer 2 Tunnel Protocol.
`Cisco-AVpair = "vpdn:ip-addresses=`*`ip-address`***`"`**	Specifies the IP address of the LNS.

Example 5-9 Configuring the RADIUS Service Profile for Tunnel Service Authorization

```
user = net1.com{
profile_id = 45
profile_cycle = 18
member = me
radius=Cisco {
check_items= [
2=cisco
}
reply_attributes= {
9,1="vpdn:tunnel-id=LAC-1"
9,1="vpdn:12tp-tunnel_password=MySecret"
9,1="vpdn:tunnel-type=12tp"
9,1="vpdn:ip-addresses=10.16.10.10"
6=5
}
}
}
```

Verifying the RADIUS Service Profile for Tunnel Service Authorization

To verify the RADIUS service profile, see your RADIUS server user documentation.

Configuring Sessions Per Tunnel Limiting in the RADIUS Service Profile

To use a RADIUS server to limit the number of sessions per tunnel, enter the following Cisco-AVpair attributes in the RADIUS service profile:

- vpdn:ip-addresses
- vpdn:ip-address-limits

Note You can configure the RADIUS server or the LAC to limit the number of sessions per tunnel. For information on using the LAC for sessions per tunnel limiting, see the "Configuring Sessions Per Tunnel Limiting on the LAC" section on page 5-12.

VPDN IP Addresses

The vpdn:ip-addresses attribute specifies the IP addresses of the LNS devices to receive the L2TP connections. It has the following syntax:

Cisco-AVpair = "vpdn:ip-addresses=*address1*[<*delimiter*>*address2*][<*delimiter*>*address3*]...**"**

The *address* argument is the IP address of the LNS.

The <*delimiter*>**,** (comma) and <*delimiter*> (space) arguments select load sharing among IP addresses.

The <*delimiter*>**/** (slash) argument groups IP addresses on the left side in higher priority than the right side.

Example 5-10 VPDN IP Addresses—RADIUS Freeware Format

In the following example, the LAC sends the:

- First PPP session through a tunnel to 10.16.1.1
- Second PPP session to 10.16.2.2
- Third PPP session to 10.16.3.3
- Fourth PPP session to 10.16.1.1

If the LAC fails to establish a tunnel with any of the IP addresses in the first group, it attempts to connect to the IP addresses in the second group (10.16.4.4 and 10.16.5.5).

```
Cisco-AVpair="vpdn:ip-addresses=10.16.1.1,10.16.2.2,10.16.3.3/10.16.4.4,10.16.5.5"
```

VPDN IP Address Limits

The vpdn:ip-address-limits attribute specifies the maximum number of sessions in each tunnel to the IP addresses listed with the attribute. It has the following syntax:

Cisco-AVpair = "vpdn:ip-address-limits=*limit1*[*limit2*][*limit3*]...**"**

The *limit* argument is the maximum number of sessions per tunnel to the corresponding IP address.

Example 5-11 VPDN IP Address Limits—RADIUS Freeware Format

```
Cisco-AVpair="vpdn:ip-address-limits=10 20 30 40 50 "
.1.1.1,10.2.2.2,10.3.3.3/10.4.4.4,10.5.5.5"
```

Note You must enter a space between the final *limit* entry and the end quotation marks.

Verifying Sessions Per Tunnel Limiting in the RADIUS Service Profile

To verify the RADIUS service profile, see the user documentation for your RADIUS server.

Configuration Example for LAC

The following example is a basic LAC configuration in which the LNS authenticates the PPP sessions.

```
Current configuration : 4882 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname c10k_mc_10005_1
!
no logging console
aaa new-model
!
!
aaa session-id common
enable password lab
!
username LAC1-1 nopassword
username LNS1-1 nopassword
no spd enable
facility-alarm intake-temperature major 49
facility-alarm intake-temperature minor 40
facility-alarm core-temperature major 53
facility-alarm core-temperature minor 45
card 1/0 1gigethernet-1
card 2/0 1oc12atm-1
card 3/0 1oc12atm-1
card 4/0 4oc3atm-1
card 5/0 1gigethernet-1
ip subnet-zero
no ip gratuitous-arps
ip host zeppelin-2 1.0.0.253
ip host zeppelin-3 1.0.0.253
!
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol pppoe
  virtual-template 1
 pppoe limit per-mac 32000
 pppoe limit per-vc 32000
!
vpdn-group LAC_1
 request-dialin
  protocol l2tp
  domain hello1
 initiate-to ip 103.1.1.2
 local name LAC1-1
 l2tp tunnel password 7 06121A2F424B05
!
!
```

```
buffers small permanent 15000
buffers middle permanent 12000
buffers large permanent 1000
!
interface Loopback1
 no ip address
!
interface FastEthernet0/0/0
 ip address 23.3.6.3 255.255.0.0
 full-duplex
!
interface GigabitEthernet1/0/0
 no ip address
 no ip mroute-cache
 negotiation auto
 hold-queue 4096 in
 hold-queue 4096 out
!
interface GigabitEthernet1/0/0.101
 encapsulation dot1Q 101
 ip address 103.1.1.1 255.255.255.0
!
interface ATM2/0/0
 no ip address
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-4
 no atm auto-configuration
 no atm ilmi-keepalive
 no atm address-registration
 no atm ilmi-enable
!
interface ATM3/0/0
 atm pppatm passive
 no ip address
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-4
 no atm auto-configuration
 no atm ilmi-keepalive
 no atm address-registration
 no atm ilmi-enable
!
interface ATM3/0/0.41101 point-to-point
 atm pppatm passive
 pvc 41/101
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41102 point-to-point
 pvc 41/102
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41103 point-to-point
 pvc 41/103
  encapsulation aal5snap
  protocol pppoe
 !
!
```

```
interface ATM3/0/0.41104 point-to-point
 pvc 41/104
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41105 point-to-point
 pvc 41/105
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41106 point-to-point
 pvc 41/106
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41107 point-to-point
 pvc 41/107
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41108 point-to-point
 pvc 41/108
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41109 point-to-point
 pvc 41/109
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41110 point-to-point
 pvc 41/110
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41111 point-to-point
 pvc 41/111
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41112 point-to-point
 pvc 41/112
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41113 point-to-point
 pvc 41/113
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41114 point-to-point
 pvc 41/114
  encapsulation aal5snap
  protocol pppoe
```

```
 !
!
interface ATM3/0/0.41115 point-to-point
 pvc 41/115
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41116 point-to-point
 pvc 41/116
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41117 point-to-point
 pvc 41/117
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41118 point-to-point
 pvc 41/118
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41119 point-to-point
 pvc 41/119
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41120 point-to-point
 pvc 41/120
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41121 point-to-point
 pvc 41/121
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41122 point-to-point
 pvc 41/122
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41123 point-to-point
 pvc 41/123
  encapsulation aal5snap
  protocol pppoe
 !
!
interface ATM3/0/0.41124 point-to-point
 pvc 41/124
  encapsulation aal5snap
  protocol pppoe
 !
!
```

```
interface ATM4/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM4/0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM4/0/2
 no ip address
 no atm ilmi-keepalive
!
interface ATM4/0/3
 no ip address
 no atm ilmi-keepalive
!
interface GigabitEthernet5/0/0
 no ip address
 negotiation auto
!
interface Virtual-Template1
 ip unnumbered Loopback1
 keepalive 30
 no peer default ip address
 ppp authentication pap
!
ip default-gateway 23.3.0.4
ip classless
ip route 1.0.0.253 255.255.255.255 23.3.0.4
no ip http server
ip pim bidir-enable
!
no cdp run
!
radius-server retransmit 3
radius-server authorization permit missing Service-Type
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
end
```

Monitoring and Maintaining LAC

To monitor and maintain the LAC, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the current configuration of the Cisco 10000 series router, acting as the LAC device. This command is useful in verifying that you successfully configured the LAC features, such as the maximum number of sessions per tunnel, the static domain name, and the LAC to RADIUS communication for tunnel service authorization

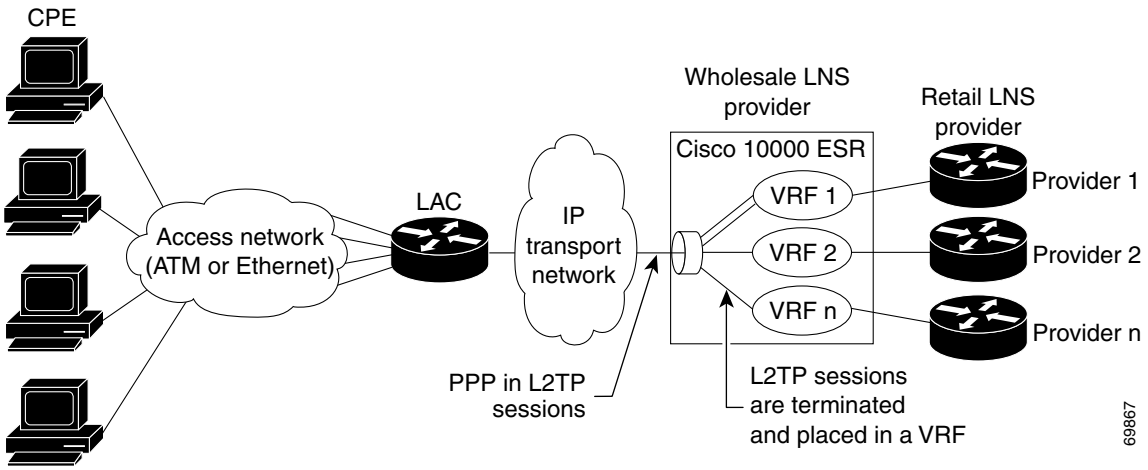
Command	Purpose
`Router# show vpdn session`	Verifies active L2TP sessions in a VPDN environment.
`Router# show vpdn tunnel`	Verifies active L2TP tunnel information in a VPDN environment.

L2TP Network Server

The Cisco 10000 series router can function as an L2TP network server (LNS). By using the managed LNS features introduced in Cisco IOS Release 12.2(4)BZ1, the Cisco 10000 series router terminates L2TP sessions from the LAC and places each session into the appropriate VRF instance based on the L2TP tunnel the session arrived in. The Cisco 10000 router then routes each session within the VRF to the destination network.

The LNS is a peer to the LAC and sits on one side of an L2TP tunnel. The LNS routes packets to and from the LAC and a destination network. Acting as the LNS, you can configure the Cisco 10000 series router to terminate the PPP sessions and route the client IP packets onto the ISP or corporate network toward their final destination (see Figure 1-1 on page 1-3). You can also configure the LNS to place the sessions in VRFs before routing the packets, as shown in Figure 5-3.

Figure 5-3 Managed LNS Topology



All of a service provider's subscribers do not share the same L2TP trunk interface. Typically, the Cisco 10000 router uses virtual local area networks (VLANs) to separate a service provider's subscriber traffic. The Cisco 10000 series router can also use permanent virtual circuits (PVCs) or a separate physical interface for each provider to separate traffic. A virtual template interface configures the user sessions in a tunnel and applies to all users in the same VRF.

The LNS feature is described in the following topics:

- Virtual Template Interface, page 5-23
- Virtual Routing and Forwarding Instance, page 5-23
- Per VRF AAA, page 5-23
- Private Servers, page 5-24
- RADIUS Attribute Screening, page 5-24
- Packet Fragmentation, page 5-24

Virtual Template Interface

The virtual template interface is a logical entity that the Cisco 10000 series router applies dynamically as needed to a connection. It is a configuration for an interface, but it is not tied to the physical interface. It is used to create and configure a virtual interface known as a virtual access interface (VAI). The VAI is cloned from the virtual template interface, used on demand, and then freed when no longer needed.

For example, when a remote user initiates a PPP session to the Cisco 10000 series router, the predefined configuration template is used to configure a VAI. The VAI is created and configured dynamically using the virtual template interface. Using AAA, RADIUS attributes can further define the VAI configuration.

The VAI uses the attributes of the virtual template to create the session, which results in a VAI that is uniquely configured for a specific user. When the user is done, the VAI goes down and the resources are freed for other client uses.

Virtual Routing and Forwarding Instance

A virtual routing and forwarding (VRF) instance includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router, such as the Cisco 10000 series router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

To configure a VRF instance, enter the **rd** command in VRF configuration submode to specify the correct route distinguisher (RD) used for the VPN. The RD extends the IP address so that you can identify the VPN to which it belongs.

Per VRF AAA

The per VRF AAA feature enables you to partition authentication, authorization, and accounting (AAA) services based on a VRF instance. To support the per VRF AAA feature, the RADIUS server must be VRF aware.

To be VRF aware, ISPs must define multiple instances of the same operational parameters and secure them to the VRF partitions. Securing AAA parameters to a VRF can be accomplished from one or more of the following sources:

- Virtual template—Used as a generic interface configuration.
- Service provider AAA server—Used to associate a remote user with a specific VPN based on the domain name. The server then provides the VPN-specific configuration for the virtual access interface that includes the IP address and port number of the customer AAA server.
- Customer VPN AAA server—Used to authenticate the remote user and to provide user-specific configurations for the virtual access interface.

For more information on the per VRF AAA feature, see the "Configuring per VRF AAA Services" section on page 5-31 and the "RADIUS Attribute Screening" section on page 16-1.

Private Servers

Private servers are servers defined within a server group. These servers have private addresses within the default server group containing all the servers. Private servers remain hidden from other groups. If you do not specify private server parameters, global configurations are used. If you do not specify global configurations, default values are used.

You configure all server operational parameters per host, per server group, or globally. Per host configurations have precedence over per server group configurations. Per server group configurations have precedence over global configurations.

RADIUS Attribute Screening

The RADIUS Attribute Screening feature allows you to configure a list of "accept" or "reject" RADIUS attributes on the Cisco 10000 series router for authorization and accounting purposes. Based on the accept or reject list you configure for a particular purpose, the Cisco 10000 series router:

- Accepts and processes all standard RADIUS attributes
- Rejects all standard RADIUS attributes

Before you configure a RADIUS accept or reject list, you must enable AAA using the **aaa new-model** command in global configuration mode. For more information, see the "Configuring RADIUS Attribute Accept or Reject Lists" section on page 5-37, the "RADIUS Attribute Screening" section on page 16-1, or see the *Cisco IOS Command Summary, Volume 2 of 3, Release 12.2*.

Packet Fragmentation

The setting of the Don't Fragment (DF) bit determines if a packet is eligible for fragmentation. If the DF bit is clear, a packet is fragmented only if it exceeds the maximum transfer unit (MTU) size. If the DF bit is set, a packet is not fragmented and instead is dropped. For packets entering an L2TP tunnel that exceed the MTU size, enter the following command in global configuration mode to configure the Cisco 10000 series router to ignore the setting of the DF bit and to fragment the packets:

```
Router(config)# [no] ip pxf ignore l2tp df-bit
```

When you activate packet fragmentation, the router clears the DF bit of packets entering all L2TP tunnels and fragments the packets, but only if the packets exceed the session MTU. Clearing the DF bit allows packets to be fragmented. If a packet enters an L2TP tunnel, but it does not exceed the MTU, the router does not clear the DF bit. Instead, the DF bit is left untouched and the router does not fragment the packet.

Tunnel Accounting

The tunnel accounting feature enhances AAA accounting by adding the ability to include tunnel-related statistics in the RADIUS information. To collect tunnel usage information, RADIUS accounting includes tunnel accounting attributes and additional tunnel accounting values for the Acct-Status-Type RADIUS attribute.

Note For more information about the RADIUS tunnel accounting attributes or the Acct-Status-Type values that support RADIUS tunnel accounting, see the "Configuring Vendor-Specific Attributes on RADIUS" section on page 5-44 or see RFC 2867.

By using the tunnel accounting feature, you can track the services that users are accessing and the amount of network resources that they are consuming. In L2TP dial-up networks, tunneling of user sessions can be done automatically as a service of the Internet service provider (ISP). This service is used to provide remote intranet access to the employees of a corporation. ISPs collect usage information about the service, which they then can use for billing purposes and for managing the network. Tunnel accounting allows dial-up usage information to be collected and stored at a central location.

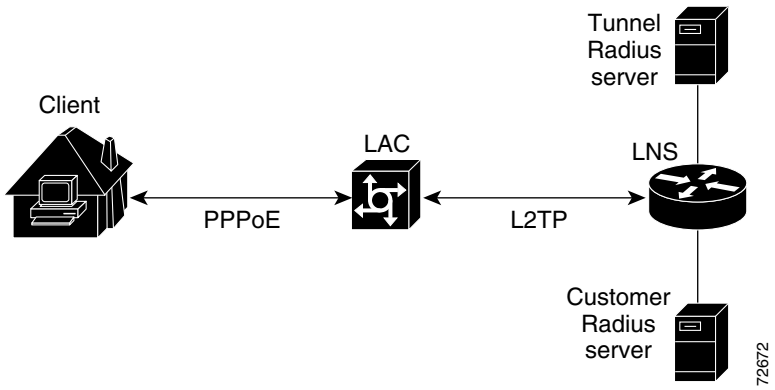
When you enable tunnel accounting on the Cisco 10000 series router, the router reports user activity to the RADIUS server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs. Accounting records are stored on the RADIUS server and can be analyzed for network management, client billing, and auditing. Corporations contracting with ISPs also receive a record of a user's resource consumption, which enables the corporation to audit its ISP billing statements.

Note For more information about AAA accounting, see the "Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Tunnel Authentication

The tunnel authentication feature verifies users before they are allowed access to the network and the network services. On the LNS, L2TP tunnel authorization and authentication can occur by using the **vpdn-group** commands configured in the local configuration. If a large number of VPDN groups is configured, maintaining the local configuration across a number of LNS devices can be difficult. To alleviate this, the Cisco 10000 series router supports the capability to do tunnel authentication using a RADIUS server.

Figure 5-4 Tunnel Authorization and Authentication



As shown in Figure 5-4, typically, a tunnel RADIUS server is used for tunnel authorization and a separate user RADIUS server is used for RADIUS tunnel authentication. The following describes the sequence of events that occur for tunnel authorization and authentication:

1. The LNS gets a Start-Control-Connection-Request (SCCRQ) and starts tunnel initialization and authorization.
2. The LNS makes an authorization request to the RADIUS server. This request includes the name of the LAC device that initiated the tunnel. The RADIUS server uses the LAC name in determining user authorization.
3. The RADIUS server determines if local or RADIUS authorization should be done. If authorization is done locally, the LNS searches the VPDN groups. If RADIUS authorization is to be done, the RADIUS server makes a RADIUS request to the LNS. This request includes the LAC host name and a hardwired password.
4. The LNS checks RADIUS attributes 90 (Tunnel-Client-Auth-ID) and 69 (Tunnel-Password). If the value in attribute 90 is inconsistent with the LAC host name or the value in attribute 69 does not match the shared secret received in the SCCRQ, the tunnel is dropped.
5. The LNS terminates the L2TP tunnel.
6. User authentication occurs either locally or by using the RADIUS server.

Note

- The Cisco 10000 series router implements tunnel authentication by using Cisco-specific RADIUS attributes. For more information about the tunnel authentication vendor-specific attributes (VSAs), see the "Configuring Vendor-Specific Attributes on RADIUS" section on page 5-44.
- For more information about AAA authentication, see the "Configuring Authentication" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Named Method Lists

To configure authentication, authorization, and accounting (AAA), you first define a named list of methods and then apply that list to various interfaces. The named method list defines the types of authentication or accounting to be performed and the sequence in which they will be performed. You must apply the method list to a specific interface before any defined authentication methods are performed. The only exception is the default method list, which is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

An authentication method list lists the methods to be queried to authenticate users. An accounting method list lists the methods used to support accounting. Method lists enable you to designate one or more security protocols to be used for authentication or accounting, thus ensuring a backup system for authentication or accounting in case the initial method fails. Cisco IOS software uses the first listed method to authenticate users or to support accounting. If that method fails to respond, the Cisco IOS software selects the next authentication or accounting method listed in the method list. This process continues until successful communication with a listed authentication or accounting method occurs, or all methods defined in the method list are exhausted.

The Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle (for example, the RADIUS server responds by denying user access), the authentication process stops and no other authentication methods are attempted.

For more information, see the "Configuring Authentication" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Framed-Route VRF Aware

The Framed-Route VRF aware feature allows you to apply static IP routes to a specific VRF table instead of the global routing table. This feature makes RADIUS Attribute 22 (Framed-Route) and a combination of Attribute 8 (Framed-IP-Address) and Attribute 9 (Framed-IP-Netmask) aware of VRF instances.

You can configure a per-user static route by using the Framed-Route attribute in any of the following ways:

- Using the Cisco **route** command
- Using the RADIUS Framed-Route attribute

Note When the PE router receives a Framed-Route attribute from the RADIUS server, the PE determines if the user is a VPN customer. If so, then the static route is implemented in the VRF routing table to which the user belongs.

- Using the RADIUS Framed-IP-Address or Framed-IP-Netmask attribute

Note The Framed-IP-Netmask attribute has the same function as the Framed-Route attribute.

Feature History for LNS

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for the LNS

To function as a LNS, the Cisco 10000 series router has the following restrictions:

- The Cisco 10000 series router does not support the configuration of L2TP tunnels over the management Fast Ethernet interface. Do not set up L2TP tunnels over this interface.
- In Cisco IOS Release 12.3(7)XI1, the output rate limited traffic on an L2TP VAI can be lower than than in previous releases due to increases in the overhead included in the policed bps rate.

  The configured police bps rate when applied to an L2TP virtual access interface includes the following 40 bytes of per packet overhead:

  - L2TP (8 bytes)
  - PPP (4 bytes)
  - Outer IP (20 bytes)
  - UDP (8 bytes)

Prerequisites for LNS

To function as an LNS, the Cisco 10000 series router has the following requirements:

- Before you configure RADIUS tunnel accounting or authentication, you must first:
  - Enable AAA on the LNS and the LAC by using the **aaa new-model** global configuration command. For more information, see the "AAA Overview" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.
  - Configure the LNS and LAC to communicate with the RADIUS server. For more information, see the "Configuring the LAC to Communicate with the RADIUS Server" section on page 5-11 and see the "Configuring RADIUS" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Note For more information, see the "Configuring Accounting," "Configuring Authentication," and "Configuring RADIUS" chapters in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Required Configuration Tasks for LNS

To configure the Cisco 10000 series router as an LNS, perform the following required configuration tasks:

- Configuring the Virtual Template Interface, page 5-29
- Configuring the LNS to Initiate and Receive L2TP Traffic, page 5-29

Note You must also configure the LAC and RADIUS server to communicate with the LNS. For more information, see the "Required Configuration Tasks for LAC" section on page 5-7 or see your RADIUS documentation.

Configuring the Virtual Template Interface

To configure a virtual template interface, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 4	`Router(config-if)# ip vrf forwarding` <*name*>	Maps the virtual template interface to a VRF routing table.
Step 5	`Router(config-if)# ip unnumbered loopback` <*number*>	Enables IP without assigning a specific IP address on the LAN.
Step 6	`Router(config-if)# ppp authentication {pap | chap | ms-chap}`	Enables PAP or CHAP authentication on the virtual template interface, which is applied to VAIs.

Configuring the LNS to Initiate and Receive L2TP Traffic

To configure the Cisco 10000 router, acting as the LNS, to initiate and receive L2TP traffic, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# vpdn enable`	Enables VPDN networking on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server (home gateway) if one is present.
Step 4	`Router(config)# vpdn-group` *group-name*	Defines a local group name for which you can assign other VPDN variables. Enters VPDN group configuration mode.
Step 5	`Router(config-vpdn)# accept-dialin`	Configures the LNS to accept tunneled PPP connections from the LAC and creates an accept-dialin VPDN subgroup.

	Command	Purpose
Step 6	`Router(config-vpdn-acc-in)# protocol 12tp`	Specifies the Layer 2 Tunnel Protocol.
Step 7	`Router(config-vpdn-acc-in)# virtual-template` *template-number*	Specifies the virtual template to be used to clone virtual access interfaces.
Step 8	`Router(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 9	`Router(config-vpdn)# terminate-from hostname` *host-name*	Specifies the host name of the remote L2TP access concentrator (LAC) that will be required when accepting a VPDN tunnel.

Example 5-12 Configuring the LNS

```
!Configures the VRF.
ip vrf vpn-1
    rd 1100:1
!
!Configures the virtual template interface and associates the VRF to it.
interface virtual-template 1
    ip vrf forwarding vpn-1
    ip unnumbered loopback
    ppp authentication chap
!
!Configures a VPDN group to ensure that all the sessions for a particular tunnel get the
same virtual template and thus the same VRF.
vpdn enable
vpdn-group 1
    accept-dialin
    protocol 12tp
    virtual-template 1
    terminate-from hostname lac1-vpn1
    local name r4-1
    12tp tunnel password 7 1511021F0725
    12tp tunnel receive-window 100
    12tp tunnel retransmit retries 7
    12tp tunnel retransmit timeout min 2
```

Optional Configuration Tasks for LNS

To configure the Cisco 10000 series router as an LNS, perform as many of the following configuration tasks as desired. All of these configuration tasks are optional.

- Configuring per VRF AAA Services, page 5-31
- Configuring a VRF on the LNS, page 5-36
- Configuring Sessions per Tunnel Limiting on the LNS, page 5-36
- Configuring RADIUS Attribute Accept or Reject Lists, page 5-37
- Configuring the LNS for RADIUS Tunnel Accounting, page 5-39
- Configuring the LNS for RADIUS Tunnel Authentication, page 5-42

Configuring per VRF AAA Services

To configure per VRF AAA services, perform the following tasks:

- Enabling AAA, page 5-31
- Configuring Private Server Parameters, page 5-31
- Configuring AAA for the VRF, page 5-32
- Configuring RADIUS-Specific Commands for the VRF, page 5-34

Note For more information about configuring AAA parameters, see the *Cisco IOS Security Configuration Guide, Release 12.2*.

Enabling AAA

To enable AAA, enter the following commands.

Note For more information, see the *Cisco IOS Command Summary, Volume 2 of 3, Release 12.2*.

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa new model`	Enables AAA.

Configuring Private Server Parameters

To configure private server operational parameters, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.

Step 3	`Router(config)# aaa group server radius` *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods.
		The *group-name* argument is the character string used to name the group.
		Note When RADIUS servers are configured in a group and the first server fails to respond, the L2TP tunnel request from the LAC might time out before the LNS fails over to the second server. To avoid this, configure the LAC with the following commands in VPDN group configuration mode:
		`l2tp tunnel retransmit initial retries 5`
		`l2tp tunnel retransmit initial timeout min 2`

	Command	Purpose
Step 4	`Router(config-sg-radius)# server-private ip-address timeout seconds retransmit retries key string`	Configures the IP address of the private RADIUS server for the group server. The *ip-address* argument specifies the IP address of the private RADIUS server host. (Optional) The *seconds* argument specifies the timeout value (1 to 1000). The *string* argument specifies the authentication and encryption key for all RADIUS communications between the Cisco 10000 series router and the RADIUS server.
Step 5	`Router(config-sg-radius)# ip vrf forwarding vrf-name`	Configures the VRF reference of the AAA RADIUS server group. The *vrf-name* argument is the name assigned to a VRF instance.

Configuring AAA for the VRF

To configure AAA for the VRF, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa authentication ppp list-name method1 [method2...]`	Specifies one or more AAA authentication methods for use on serial interfaces running PPP. The *list-name* argument is a character string used to name the list of authentication methods tried when a user logs in. The *method1*[*method2...*] argument is at least one of the following keywords: • **if-needed**—Does not authenticate if user has already been authenticated on a TTY line. • **local**—Uses the local username database for authentication. • **local-case**—Uses case-sensitive local username authentication. • **none**—Uses no authentication. • **group radius**—Uses the list of all RADIUS servers for authentication. • **group** *group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command.

	Command	Purpose
Step 4	`Router(config)# aaa authorization network list-name method1 [method2...]`	Sets parameters that restrict user access to a network. The *list-name* argument is a character string used to name the list of authentication methods tried when a user logs in. The *method1*[*method2...*] argument is at least one of the following keywords: • **group radius**—Uses the list of all RADIUS servers for authentication. • **group** *group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command. • **if-authenticated**—Succeeds if user has been successfully authenticated. • **local**—Uses the local username database for authentication. • **none**—Uses no authentication.
Step 5	`Router(config)# aaa accounting {system default [vrf vrf-name] | network {default | none | start-stop | stop-only | wait-start} group group-name`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. The **system default** keyword performs accounting for all system-level events not associated with users, such as reloads. The **vrf** *vrf-name* keyword and argument specify a VRF configuration. The **network** keyword runs accounting for all network-related service requests. The **default** keyword specifies the default accounting list: • **none**—No accounting. • **start-stop**—Record stop and start without waiting. • **stop-only**—Record stop when service terminates. • **wait-start**—Record stop and start after start-record commit. The **group** *group-name* keyword and argument use a subset of RADIUS servers for accounting as defined by the server **group** *group-name*.
Step 6	`Router(config)# aaa accounting delay-start vrf vrf-name`	Delays generation of the start accounting records until the user IP address is established. The **vrf** *vrf-name* keyword and argument enables the specification on a per VRF basis.
Step 7	`Router(config)# aaa accounting send stop-record authentication failure vrf vrf-name`	Generates accounting stop records for users who fail to authenticate at login or during session negotiation. The **vrf** *vrf-name* keyword and argument enables the specification on a per VRF basis.

Configuring RADIUS-Specific Commands for the VRF

To configure AAA global RADIUS-specific commands for the VRF definition, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# interface virtual-template number`	Configures a virtual template interface and enters interface configuration mode.
Step 4	`Router(config-if)# ip vrf forwarding vrf-name`	Associates a VRF instance with a virtual template interface. The *vrf-name* argument is the name assigned to a VRF.
Step 5	`Router(config-if)# ppp authentication {protocol1 [protocol2...]} list-name`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface. The *protocol1*[*protocol2*...] argument specifies at least one of the following keywords: • **chap**—Enables CHAP on a serial interface. • **ms-chap**—Enables Microsoft's version of CHAP (MS-CHAP) on a serial interface. • **pap**—Enables PAP on a serial interface. The *list-name* argument (optional) specifies the name of a list of methods of authentication to use. This is the same name you specified in step 4 of the "Configuring AAA for the VRF" section on page 5-32. If no list name is specified, the system uses the default. Create the list by using the **aaa authentication ppp** command.
Step 6	`Router(config-if)# ppp authorization list-name`	Enables AAA authorization on the selected interface. The *list-name* argument (optional) specifies the name of a list of authorization methods to use. If no list name is specified, the system uses the default. Create the list by using the **aaa authorization** command.
Step 7	`Router(config-if)# ppp accounting list-name`	Enables AAA accounting services on the selected interface.
Step 8	`Router(config-if)# exit`	Exits interface configuration mode.
Step 9	`Router(config)# ip radius source-interface subinterface-name vrf vrf-name`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets and enables the specification on a per VRF basis. The *subinterface-name* argument specifies the name of the interface that RADIUS uses for all of its outgoing packets. The **vrf** *vrf-name* keyword and argument specify the per VRF configuration.

	Command	Purpose
Step 10	`Router(config)# radius-server attribute 44 include-in-access-req vrf vrf-name`	Sends RADIUS attribute 44 in access request packets before user authentication and enables the specification on a per VRF basis. The **vrf** *vrf-name* keyword and argument specify the per VRF configuration.
Step 11	`Router(config)# radius-server domain-stripping vrf vrf-name`	(Optional) Enables VRF-aware domain-stripping. The **vrf** *vrf-name* keyword and argument specify the per VRF configuration.

Verifying and Troubleshooting per VRF AAA

To verify and troubleshoot the per VRF AAA feature, enter the following commands in privileged EXEC mode.

Note Due to the large output of some of the commands, many events are not displayed on the console. Instead, the messages are logged to a console log file. To limit the rate that the Cisco 10000 series router logs system messages, enter the **logging rate-limit** command. For more information, see the "Troubleshooting and Fault Management Commands in the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.

Command	Purpose
`Router# show ip route vrf vrf-name`	Displays the IP routing table associated with a VRF.
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authorization`	Displays information on AAA authorization.
`Router# debug ppp negotiation`	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router# debug radius`	Displays information associated with RADIUS.
`Router# debug vpdn event`	Displays L2TP errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router# debug vpdn error`	Displays debug traces for VPN.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Configuring a VRF on the LNS

To configure a VRF, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 4	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.

For more information about configuring a VRF, see the "Configuring Multiprotocol Label Switching chapter in the *Cisco IOS Switching Services Configuration Guide, Release 12.2*.

Configuring Sessions per Tunnel Limiting on the LNS

To limit the number of sessions per tunnel without using a RADIUS server, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# vpdn-group group-name`	Defines a local group name for which you can assign other VPDN variables. Enters VPDN group configuration mode.
Step 4	`Router(config-vpdn)# accept-dialin`	Configures the LNS to accept tunneled PPP connections from the LAC and creates an accept-dialin VPDN subgroup.
Step 5	`Router(config-vpdn-acc-in)# protocol 12tp`	Specifies the Layer 2 Tunnel Protocol.
Step 6	`Router(config-vpdn-acc-in)# virtual-template template-number`	Specifies the virtual template to be used to clone virtual access interfaces.
Step 7	`Router(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 8	`Router(config-vpdn)# terminate-from hostname host-name`	Specifies the host name of the remote L2TP access concentrator (LAC) that is required when accepting a VPDN tunnel.
Step 9	`Router(config-vpdn)# session-limit limit-number`	Specifies the maximum number of sessions per tunnel.

Verifying Sessions per Tunnel Limiting on the LNS

To verify sessions per tunnel limiting on the LNS, enter the following commands:

Command	Purpose
`Router# show running-config`	Displays the current router configuration. Check the output to verify that you successfully configured the maximum number of sessions per tunnel.
`Router# show vpdn tunnel`	Displays information about all active L2TP tunnels in summary-style format. Check the output to verify that the number of displayed sessions does not exceed your configured limit.

Configuring RADIUS Attribute Accept or Reject Lists

To configure a RADIUS attribute accept or reject list for authorization or accounting, enter the following commands:

	Command	Purpose
Step 1	`Router> enable`	Enters privileged EXEC mode.
Step 2	`Router# config terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa authentication ppp default group` *group-name*	Specifies one or more AAA authentication methods for use on serial interfaces running PPP.
Step 4	`Router(config)# aaa authorization network default group` *group-name*	Sets parameters that restrict network access to the user.
Step 5	`Router(config)# aaa group server radius` *group-name*	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server group configuration mode.
Step 6	`Router(config-sg-radius)# server-private` *ip-address* `timeout` *seconds* `retransmit` *retries* `key` *string*	Configures the IP address of the private RADIUS server for the group server. The *ip-address* argument specifies the IP address of the private RADIUS server host. (Optional) The *seconds* argument specifies the timeout value (1 to 1000). The *string* argument specifies the authentication and encryption key for all RADIUS communications between the Cisco 10000 series router and the RADIUS server.
Step 7	`Router(config-sg-radius)# authorization [accept | reject]` *listname* and/or `Router(config-sg-radius)# accounting [accept | reject]` *listname*	Specifies a filter for the attributes that are returned in an Access-Accept packet from the RADIUS server. Specifies a filter for the attributes that are to be sent to the RADIUS server in an accounting request. The **accept** keyword indicates that all attributes will be rejected except the attributes specified in the *listname* argument. The **reject** keyword indicates that all attributes will be accepted except for the attributes specified in the *listname* argument and all standard attributes.

	Command	Purpose
Step 8	`Router(config-sg-radius)# exit`	Exits server-group configuration mode.
Step 9	`Router(config)# radius-server attribute list` *listname*	Defines the list name given to the set of attributes defined using the **attribute** command. Define the *listname* argument to be the same as you defined it in step 5.
Step 10	`Router(config-sg-radius)# attribute` *value1* `[`*value2* `[`*value3...*`]]`	Adds attributes to the configured accept or reject list. You can use this command multiple times to add attributes to an accept or reject list.

Verifying RADIUS Attribute Accept or Reject Lists

To verify an accept or reject list, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authentication`	Displays information on AAA authentication.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Configuring the LNS for RADIUS Tunnel Accounting

To configure the LNS for RADIUS tunnel accounting, perform the following required configuration tasks:

- Configuring AAA Accounting Using Named Method Lists, page 5-39
- Configuring RADIUS for Tunnel Accounting, page 5-39

Configuring AAA Accounting Using Named Method Lists

To configure AAA accounting using named method lists, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [method1 [method2...]]`	Creates an accounting method list and enables accounting. The *list-name* argument is a character string used to name the list you are creating.

Step 2	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]`
	
	`or`
	
	`Router(config)# interface interface-type interface-number`
		Enters the line configuration mode for the line to which you want to apply the accounting method list.
		
		Enters the interface configuration mode for the interface to which you want to apply the accounting method list.

Step 3	`Router(config-line)# accounting {arap | commands level | connection | exec} {default | list-name}`
	
	`or`
	
	`Router(config-if)# ppp accounting {default | list-name}`
		Applies the accounting method list to a line or a set of lines.
		
		Applies the accounting method list to an interface.

Note System accounting does not use named method lists. For system accounting you can define only the default method list. For more information, see the "Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Configuring RADIUS for Tunnel Accounting

Cisco IOS Release 12.2(15)BX enhances the AAA accounting feature by adding the ability to include tunnel-related statistics in the RADIUS information. To collect tunnel usage information, you must configure the following attributes on the RADIUS server:

- Acct-Tunnel-Connection—Specifies the identifier assigned to the tunnel session. This attribute and the Tunnel-Client-Endpoint and Tunnel-Server-Endpoint attributes provide a way to uniquely identify a tunnel session for auditing purposes.
- Acct-Tunnel-Packets-Lost—Specifies the number of packets lost on a given link.

Table 5-1 describes the values for the Acct-Status-Type attribute that support tunnel accounting on the RADIUS server.

Table 5-1 Acct-Status-Type Values for RADIUS Tunnel Accounting

Acct-Status-Type Values	Value	Description
Tunnel-Start	9	Marks the establishment of a tunnel with another device.
Tunnel-Stop	10	Marks the destruction of a tunnel to or from another device.
Tunnel-Reject	11	Marks the rejection of the establishment of a tunnel with another device.
Tunnel-Link-Start	12	Marks the creation of a tunnel link within an L2TP tunnel that carries multiple links.
Tunnel-Link-Stop	13	Marks the destruction of a tunnel link within an L2TP tunnel that carries multiple links.
Tunnel-Link-Reject	14	Marks the rejection of the establishment of a new link in an existing tunnel.

Example 5-13 is an example of Tunnel-Start accounting record sent by the LNS to the RADIUS server.

Example 5-13 Tunnel-Start Accounting Record

```
User-Name = LNS1/LAC1
NAS-IP-Address = 23.1.2.10
Service-Type = Framed
Framed-Protocol = PPP
Tunnel-Type_tag0 = L2TP
Tunnel-Medium-Type_tag0 = IPv4
Tunnel-Client-Endpoint_tag0 = 10.2.2.1
Tunnel-Server-Endpoint_tag0 = 10.2.2.2
Acct-Status-Type = Tunnel-Start
Acct-Delay-Time = 0
Acct-Session-Id = 00000B3D
Acct-Authentic = RADIUS
Tunnel-Client-Auth-ID_tag0 = LAC1
Tunnel-Server-Auth-ID_tag0 = LNS1
Acct-Tunnel-Connection = 63708/13441
```

Example 5-14 is an example of a Tunnel-Stop accounting record sent by the LNS to the RADIUS server.

Example 5-14 Tunnel-Stop Accounting Record

```
User-Name = LNS1/LAC1
NAS-IP-Address = 23.1.2.10
Service-Type = Framed
Framed-Protocol = PPP
Ascend-Multilink-ID = 2877
Ascend-PreSession-Time = 0
Tunnel-Type_tag0 = L2TP
Tunnel-Medium-Type_tag0 = IPv4
Tunnel-Client-Endpoint_tag0 = 10.2.2.1
Tunnel-Server-Endpoint_tag0 = 10.2.2.2
Ascend-Pre-Input-Packets = 0
Ascend-Pre-Input-Octets = 0
```

```
Acct-Status-Type = Tunnel-Stop
Acct-Delay-Time = 0
Acct-Input-Octets = 108276
Acct-Output-Octets = 65986
Acct-Session-Id = 00000B3D
Acct-Authentic = RADIUS
Acct-Session-Time = 57
Acct-Input-Packets = 2578
Acct-Output-Packets = 2823
Acct-Terminate-Cause = NAS Error
Acct-Multi-Session-Id = 00000B3D
Tunnel-Client-Auth-ID_tag0 = LAC1
Tunnel-Server-Auth-ID_tag0 = LNS1
Ascend-Connect-Progress = Call-Up
Acct-Tunnel-Connection = 63708/13441
Ascend-Disconnect-Cause = No-Reason
Acct-Tunnel-Packets-Lost = 0
Ascend-Pre-Output-Octets = 0
Ascend-Pre-Output-Packets = 0
```

For more information about the RADIUS tunnel accounting attributes or the Acct-Status-Type values that support RADIUS tunnel accounting, see RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*.

For information about RADIUS accounting attributes supported on the Cisco 10000 series router, see Appendix A, "RADIUS Attributes".

For information about RADIUS attributes, see the "RADIUS Attributes" appendix in the *Cisco IOS Security Configuration Guide, Release 12.2*.

For more information on configuring RADIUS, see your RADIUS user documentation.

Configuring Optional RADIUS Tunnel Accounting Features

To configure RADIUS tunnel accounting, you can also perform any of the following optional configuration tasks:

- Suppressing Generation of Accounting Records for Null Username Sessions
- Generating Interim Accounting Records
- Generating Accounting Records for Failed Login or Session
- Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records
- Configuring AAA Resource Failure Stop Accounting
- Configuring AAA Resource Accounting for Start-Stop Records
- Configuring AAA Broadcast Accounting
- Configuring AAA Resource Failure Stop Accounting
- Configuring AAA Session MIB
- Monitoring Accounting
- Troubleshooting Accounting

Note For more information, see the "Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Configuring the LNS for RADIUS Tunnel Authentication

To configure the LNS for RADIUS tunnel authentication, perform the following required configuration tasks:

- Configuring RADIUS Tunnel Authentication Method Lists on the LNS, page 5-42
- Configuring AAA Authentication Methods, page 5-43
- Configuring Vendor-Specific Attributes on RADIUS, page 5-44

Note Cisco 10000 series router supports L2TP tunnel authorization, however, RADIUS does not provide attributes for such parameter values as L2TP tunnel timeouts, L2TP tunnel hello intervals, and L2TP tunnel receive window size. When the Cisco 10000 series router does not receive a RADIUS attribute for a parameter, the router uses the default value.

Configuring RADIUS Tunnel Authentication Method Lists on the LNS

To configure method lists on the LNS for RADIUS tunnel authentication, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization network list-name method1 [method2...]`	Sets parameters that restrict user access to a network. The *list-name* argument is a character string used to name the list of authentication methods tried when a user logs in. The *method1*[*method2*...] argument is at least one of the following keywords: • **group radius**—Uses the list of all RADIUS servers for authentication. • **group** *group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command. • **if-authenticated**—Succeeds if the user has been successfully authenticated. • **local**—Uses the local username database for authentication. • **none**—Uses no authentication. **Note** The method list is only for VPDN tunnel authorization and termination, not for domain and DNIS authorization. Therefore, the method list applies only on the tunnel terminator device: the LAC for dialout sessions and the LNS for dialin sessions.
Step 2	`Router(config)# vpdn tunnel authorization network <method list name>`	Specifies the AAA method list to use for VPDN remote tunnel hostname-based authorization. If you do not specify a method list (including a default method list) by using the **vpdn tunnel authorization network** command, local authorization occurs by using the local VPDN group configuration.

	Command	Purpose
Step 3	`Router(config)# vpdn tunnel authorization virtual-template <vtemplate num>`	Specifies the default virtual template interface used to clone a virtual access interface (VAI). If you do not specify a virtual template interface in the local VPDN group configuration or in a remote RADIUS configuration, then this default virtual template interface is used. **Note** The **vpdn tunnel authorization virtual-template** command is only applicable on the LNS.
Step 4	`Router(config)# vpdn tunnel authorization password <dummy password>`	Specifies the password to use for the RADIUS authorization request to retrieve the tunnel configuration based on the remote tunnel hostname. By default, the password is *cisco*, but you can configure a different password. **Note** The **vpdn tunnel authorization password** command is applicable on both the LAC and LNS.

Configuring AAA Authentication Methods

To configure AAA authentication methods, do the following:

Step 1 Enable AAA using the **aaa new-model** global configuration command. For more information, see the "AAA Overview" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Step 2 Configure RADIUS security protocol parameters. For more information about RADIUS, see the "Configuring RADIUS" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Step 3 Define the authentication method lists using the **aaa authentication** command.

Step 4 Apply the authentication method lists to an interface, a line, or a set of lines as required.

The Configuring Authentication" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2* describes how to configure the following authentication methods:

- Configuring Login Authentication Using AAA
- Configuring PPP Authentication Using AAA
- Configuring AAA Scalability for PPP Requests
- Configuring ARAP Authentication Using AAA
- Configuring NASI Authentication Using AAA
- Specifying the Amount of Time for Login Input
- Enabling Password Protection at the Privileged Level
- Changing the Text Displayed at the Password Prompt
- Configuring Message Banners for AAA Authentication
- Configuring AAA Packet of Disconnect
- Enabling Double Authentication
- Enabling Automated Double Authentication

Configuring Vendor-Specific Attributes on RADIUS

Cisco IOS Release 12.2(15)BX adds Cisco-specific VPDN RADIUS attributes to support RADIUS tunnel authentication. To configure the RADIUS server for tunnel authentication, you must configure the following vendor-specific attributes (VSAs) on the RADIUS server:

- vpdn-vtemplate—Specifies the virtual template number to use for cloning on the LNS. This attribute corresponds to the virtual template associated with the local VPDN group on the LNS. This attribute is not required if you used the **vpdn tunnel authorization virtual-template** <*vtemplate num*> command on the LNS to configure a default virtual template to use for cloning.

  ```
  Cisco:Cisco-Avpair = "vpdn:vpdn-vtemplate = <vtemplate number>"
  ```

- dout-dialer—Specifies the LAC dialer to use on the LAC for a dialout configuration.

  ```
  Cisco:Cisco-Avpair = "vpdn:dout-dialer = <LAC dialer number>"
  ```

- Service-Type—Specifies an outbound or inbound service type. In the tunnel authorization request, the LNS sets the Service-Type attribute to Outbound. Therefore, in the RADIUS configuration you must also configure an Outbound Service-Type.

  ```
  Service-Type = Outbound
  ```

Note

- For information about RADIUS attributes supported on the Cisco 10000 series router, see Appendix A, "RADIUS Attributes" or see the "RADIUS Attributes" appendix in the *Cisco IOS Security Configuration Guide, Release 12.2*.
- For more information about configuring RADIUS, see your RADIUS user documentation.

Example 5-15 is a RADIUS configuration that allows the LNS to terminate L2TP tunnels from a LAC. In this configuration, VirtualTemplate10 is used to clone a virtual access interface (VAI) on the LNS.

Example 5-15 Configuring RADIUS for LNS Termination of L2TP Tunnels from a LAC

```
myLACname       Password = "cisco"
    Service-Type = Outbound,
    Tunnel-Type = :0:l@TP,
    Tunnel-Medium-Type = :o:IP,
    Tunnel-Client-Auth-ID = :0:"myLACname",
    Tunnel-Password = :0:"mytunnelpassword",
    Cisco:Cisco-Avpair = "vpdn:vpdn-vtemplate=10"
```

Example 5-16 is an LNS configuration that supports RADIUS tunnel authentication. In this configuration, a RADIUS server group is defined using the **aaa group server radius VPDN-Group** command. The **aaa authorization network mymethodlist group VPDN-Group** command queries RADIUS for network authorization.

Example 5-16 Configuring the LNS to Support RADIUS Tunnel Authentication

```
aaa group server radius VPDN-Group
    server 64.102.48.91 auth-port 1645 acct-port 1646
aaa authorization network mymethodlist group VPDN-Group
vpdn tunnel authorization network mymethodlist
vpdn tunnel authorization virtual-template 10
```

Configuration Examples for LNS

This section provides example configurations for the following features:

Managed LNS Configuration Example

Example 5-17 is an example of how to configure the Managed LNS features on the Cisco 10000 series router. In this example, the Cisco 10000 series router terminates the tunnel from the LAC and associates the VRFs with the interfaces and the virtual template interfaces. This configuration also configures RADIUS attribute screening and AAA accounting for the VRFs.

Example 5-17 Configuring Managed LNS on the Cisco 10000 Series Router

```
!Enables AAA.
aaa new-model
!
!Configures private server parameters.
aaa group server radius vpn1
 server-private 192.168.1.128 auth-port 1645 acct-port 1646 key cisco
 server-private 192.168.2.128 auth-port 1645 acct-port 1646 timeout 10 retransmit 3 key
!Configures RADIUS attribute screening.
cisco1
 authorization reject vpn1-autho-list
 accounting reject vpn1-account-list
 ip vrf forwarding vpn1
!
!Configures private server parameters.
aaa group server radius vpn2
 server-private 192.168.1.128 auth-port 1645 acct-port 1646 key cisco
 server-private 192.168.2.128 auth-port 1645 acct-port 1646 timeout 10 retransmit 3 key
cisco1
 ip vrf forwarding vpn2
!
!Configures AAA accounting for the VRFs.
aaa authentication ppp vpn1 group vpn1
aaa authentication ppp vpn2 group vpn2
aaa authorization network vpn1 group vpn1
aaa authorization network vpn2 group vpn2
aaa accounting update periodic 1
aaa accounting network vpn1 start-stop group vpn1
aaa accounting network vpn2 start-stop group vpn2
aaa accounting system default vrf vpn1 start-stop group vpn1
aaa accounting system default vrf vpn2 start-stop group vpn2
aaa session-id common
!
!Configures the VRFs.
ip vrf vpn1
 rd 1100:1
!
ip vrf vpn2
 rd 1100:2
vpdn enable
!
!Terminates the tunnel from the LAC.
vpdn-group 1
```

```
 accept-dialin
 protocol l2tp
 virtual-template 1
 terminate-from hostname lac1-vpn1
 local name r4-1
 lcp renegotiation on-mismatch
 l2tp tunnel password 7 1511021F0725
 l2tp tunnel receive-window 100
 l2tp tunnel retransmit retries 7
 l2tp tunnel retransmit timeout min 2
!
!Terminates the tunnel from the LAC.
vpdn-group 2
 accept-dialin
  protocol l2tp
  virtual-template 2
 terminate-from hostname lac1-vpn2
 local name r4-2
 lcp renegotiation on-mismatch
 l2tp tunnel password 7 121A0C041104
 l2tp tunnel receive-window 100
 l2tp tunnel retransmit retries 7
 l2tp tunnel retransmit timeout min 2
!
!
!Associates the VRF with the interface.
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.1.1.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vpn2
 ip address 10.1.2.1 255.255.255.255
!
interface FastEthernet0/0/0
 no ip address
 shutdown
!
!Configures the interface used to connect to the LAC.
interface GigabitEthernet6/0/0
 ip address 10.1.1.45 255.255.255.0
 negotiation auto
!
interface GigabitEthernet7/0/0
 no ip address
 negotiation auto
!
!Associates the VRF with the interface.
interface GigabitEthernet7/0/0.1
 encapsulation dot1Q 11
 ip vrf forwarding vpn1
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet7/0/0.2
 encapsulation dot1Q 12
 ip vrf forwarding vpn2
 ip address 192.168.2.1 255.255.255.0
!
!Associates the VRF with the virtual template interface.
interface Virtual-Template1
 ip vrf forwarding vpn1
 ip unnumbered Loopback1
 no peer default ip address
 ppp authentication chap vpn1
```

```
 ppp authorization vpn1
 ppp accounting vpn1
!
!Associates the VRF with the virtual template interface.
interface Virtual-Template2
 ip vrf forwarding vpn2
 ip unnumbered Loopback2
 no peer default ip address
 ppp authentication chap vpn2
 ppp authorization vpn2
 ppp accounting vpn2
!
!Enters the VRFs in the routing table.
ip classless
ip route vrf vpn1 192.168.4.2 255.255.255.0 192.168.5.3
ip route vrf vpn2 192.168.4.2 255.255.255.0 192.168.5.4
no ip http server
ip pim bidir-enable
!
!Configures RADIUS-specific command for the VRF to force RADIUS to use the IP address of a
!specified interface for all outgoing RADIUS packets.
ip radius source-interface GigabitEthernet7/0/0.1 vrf vpn1
ip radius source-interface GigabitEthernet7/0/0.2 vrf vpn2
no cdp run
!
!radius-server retransmit is on by default and cannot be removed.
radius-server retransmit 3
!Configures optional features such as domain-name stripping and RADIUS attribute filter.
radius-server domain-stripping vrf vpn1
radius-server domain-stripping vrf vpn2
radius-server attribute 44 include-in-access-req vrf vpn1
radius-server attribute 44 include-in-access-req vrf vpn2
radius-server attribute list vpn1-autho-list
 attribute 26,200-220
!
radius-server attribute list vpn1-account-list
 attribute 60-70
!
```

Tunnel Accounting Configuration Examples

This section provides the following configuration examples:

- LNS Tunnel Accounting Configuration Example, page 5-48
- RADIUS Tunnel Accounting Records, page 5-49

LNS Tunnel Accounting Configuration Example

Example 5-18 shows how to configure the LNS to send tunnel accounting records to the RADIUS server.

Example 5-18 Configuring the LNS for Tunnel Accounting

```
aaa new-model
!
!
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
aaa session-id common
enable secret 5 $1$ftf.$wE6Q5Yv6hmQiwL9pizPCg1
!
username ENT_LNS password 0 tunnelpass
username user1@cisco.com password 0 lab
username user2@cisco.com password 0 lab
spe 1/0 1/7
firmware location system:/ucode/mica_port_firmware
spe 2/0 2/9
firmware location system:/ucode/mica_port_firmware
!
!
resource-pool disable
clock timezone est 2
!
ip subnet-zero
no ip domain-lookup
ip host CALLGEN-SECURITY-V2 10.24.80.28 10.47.0.0
ip host dirt 172.16.1.129
!
vpdn enable
vpdn tunnel accounting network m1
vpdn session accounting network m1
!
vpdn-group 1
accept-dialin
protocol l2tp
virtual-template 1
terminate-from hostname ISP_LAC
local name ENT_LNS
!
isdn switch-type primary-5ess
!
!
fax interface-type modem
mta receive maximum-recipients 0
!
interface Loopback0
ip address 172.16.0.101 255.255.255.0
!
interface Loopback1
ip address 192.168.0.101 255.255.255.0
!
interface Ethernet0
ip address 10.1.26.71 255.255.255.0
no ip mroute-cache
no cdp enable
!
interface Virtual-Template1
ip unnumbered Loopback0
peer default ip address pool vpdn-pool1
ppp authentication chap
```

```
!
interface Virtual-Template2
ip unnumbered Loopback1
peer default ip address pool vpdn-pool2
ppp authentication chap
!
interface FastEthernet0
no ip address
no ip mroute-cache
shutdown
duplex auto
speed auto
no cdp enable
!
ip local pool vpdn-pool1 172.16.5.1 172.16.128.100
ip local pool vpdn-pool2 10.0.0.1 10.0.0.100
ip default-gateway 10.1.26.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.26.254
ip route 192.168.1.2 255.255.255.255 10.1.26.254
no ip http server
ip pim bidir-enable
!
!
dialer-list 1 protocol ip permit
no cdp run
!
!
radius-server host 172.16.192.80 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
```

RADIUS Tunnel Accounting Records

Example 5-19 and Example 5-20 show RADIUS tunnel accounting record types.

Example 5-19 RADIUS Tunnel Accounting Record

```
User-Name = gomer1@hello101
NAS-IP-Address = 23.1.2.10
NAS-Port = 550
Service-Type = Framed
Framed-Protocol = PPP
Ascend-Multilink-ID = 2877
Tunnel-Type_tag0 = L2TP
Tunnel-Medium-Type_tag0 = IPv4
Tunnel-Client-Endpoint_tag0 = 10.2.2.1
Tunnel-Server-Endpoint_tag0 = 10.2.2.2
Acct-Status-Type = Tunnel-Link-Start
Acct-Delay-Time = 0
Acct-Session-Id = 00000B42
Acct-Authentic = RADIUS
Acct-Multi-Session-Id = 00000B3D
Tunnel-Client-Auth-ID_tag0 = LAC1
Tunnel-Server-Auth-ID_tag0 = LNS1
NAS-Port-Type = Virtual
Acct-Tunnel-Connection = 1088401809
```

Example 5-20 RADIUS Tunnel Accounting Record

```
Wed, 15 Jan 2003 16:34:27
User-Name = gomer1@hello101
NAS-IP-Address = 23.1.2.10
NAS-Port = 550
Service-Type = Framed
Framed-Protocol = PPP
Ascend-Multilink-ID = 2877
Ascend-PreSession-Time = 0
Tunnel-Type_tag0 = L2TP
Tunnel-Medium-Type_tag0 = IPv4
Tunnel-Client-Endpoint_tag0 = 10.2.2.1
Tunnel-Server-Endpoint_tag0 = 10.2.2.2
Ascend-Pre-Input-Packets = 0
Ascend-Pre-Input-Octets = 0
Acct-Status-Type = Tunnel-Link-Stop
Acct-Delay-Time = 0
Acct-Input-Octets = 462
Acct-Output-Octets = 293
Acct-Session-Id = 00000B42
Acct-Authentic = RADIUS
Acct-Session-Time = 45
Acct-Input-Packets = 11
Acct-Output-Packets = 12
Acct-Terminate-Cause = User Request
Acct-Multi-Session-Id = 00000B3D
Acct-Link-Count = 250
Tunnel-Client-Auth-ID_tag0 = LAC1
Tunnel-Server-Auth-ID_tag0 = LNS1
Ascend-Connect-Progress = LAN-Session-Up
NAS-Port-Type = Virtual
Acct-Tunnel-Connection = 1088401809
Ascend-Disconnect-Cause = PPP-Rcv-Terminate-Req
Ascend-Num-In-Multilink = 250
Acct-Tunnel-Packets-Lost = 0
Ascend-Pre-Output-Octets = 0
Ascend-Pre-Output-Packets = 0
```

Note For additional accounting examples, see the "Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide, Release 12.2*.

Tunnel Authentication Configuration Examples

This section provides the following tunnel authentication configuration examples:

- LNS Configuration to Support RADIUS Tunnel Authentication, page 5-51
- RADIUS Configuration to Support Tunnel Authentication, page 5-51

LNS Configuration to Support RADIUS Tunnel Authentication

The following example is an LNS configuration that supports RADIUS tunnel authentication. In this configuration, a RADIUS server group is defined by using the **aaa group server radius VPDN-Group** command. The **aaa authorization network mymethodlist group VPDN-Group** command queries RADIUS for network authorization.

```
aaa group server radius VPDN-Group
server 64.102.48.91 auth-port 1645 acct-port 1646
aaa authorization network mymethodlist group VPDN-Group
vpdn tunnel authorization network mymethodlist
vpdn tunnel authorization virtual-template 10
```

RADIUS Configuration to Support Tunnel Authentication

The following example is a RADIUS configuration that allows the LNS to terminate L2TP tunnels from a LAC. In this configuration, *VirtualTemplate10* is used to clone a VAI on the LNS.

```
myLACname Password = "cisco"
Service-Type = Outbound,
Tunnel-Type = :0:1@TP,
Tunnel-Medium-Type = :o:IP,
Tunnel-Client-Auth-ID = :0:"myLACname",
Tunnel-Password = :0:"mytunnelpassword",
Cisco:Cisco-Avpair = "vpdn:vpdn-vtemplate=10"
```

Note For additional authentication examples, see the "Configuring Authentication" chapter in the *Cisco IOS Security Configure Guide, Release 12.2*.

Monitoring and Maintaining LNS

To monitor and maintain the features configured on the LNS, enter the following commands in privileged EXEC mode:

Command	Purpose
Router# **show accounting**	Displays accounting records for users currently logged in. Displays active accountable events on the network and helps collect information in the event of a data loss on the accounting server.
Router# **show interfaces virtual-access** *number* [**configuration**]	Displays status, traffic data, and configuration information about the virtual access interface you specify.
Router# **show ip route vrf** *vrf-name*	Displays the IP routing table associated with a VRF.
Router# **show radius statistics**	Displays the RADIUS statistics for accounting and authentication packets.
Router# **show vpdn**	Displays all tunnel and session information for all active sessions and tunnels.
Router# **show vpdn session**	Displays information about active L2TP sessions in a virtual private dialup network (VPDN).
Router# **show vpdn session all username** *username*	Displays statistics about all active L2TP tunnels for the username you specify.

Command	Purpose
`Router# show vpdn tunnel`	Displays information about all active L2TP tunnels in a VPDN.
`Router# show vpdn tunnel all`	Displays information about all active L2TP tunnels.
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authorization`	Displays information on AAA authorization.
`Router# debug ppp chap`	Displays authentication protocol messages for Challenge Authentication Protocol (CHAP) packet exchanges. This command is useful when a CHAP authentication failure occurs due to a configuration mismatch between devices. Verifying and correcting any username and password mismatch resolves the problem.
`Router# debug ppp negotiation`	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router# debug ppp negotiation chap`	Used to decipher a CHAP negotiation problem due to a connectivity problem between a Cisco and non-Cisco device.
`Router# debug radius`	Displays information associated with RADIUS.
`Router# debug vpdn events`	Displays L2TP errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router# debug vpdn errors`	Displays errors that prevent a tunnel from being established or errors that cause an established tunnel to be closed.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

CHAPTER 6

Configuring PPPoE over Ethernet and IEEE 802.1Q VLAN

The Cisco 10000 series router allows the tunneling and termination of PPP sessions over Ethernet links. The PPPoE over Ethernet interface (PPPoEoE) feature enables the Cisco 10000 series router to tunnel and terminate Ethernet PPP sessions over Ethernet links. The PPPoE over IEEE 802.1Q VLANs feature enables the router to tunnel and terminate Ethernet PPP sessions across VLAN links. IEEE 802.1Q encapsulation is used to interconnect a VLAN-capable router with another VLAN-capable networking device. The packets on the 802.1Q link contain a standard Ethernet frame and the VLAN information associated with that frame.

This chapter describes the following features:

For more information, see the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide* and the *VLAN Range, Release 12.2(13)T* feature guide.

PPPoE over Ethernet

The PPPoE over Ethernet feature provides the ability to connect a network of hosts over a simple bridging access device to a remote Access Concentrator. The Cisco 10000 series router supports PPPoE over Ethernet sessions to enable multiple hosts on a shared Ethernet interface to open PPP sessions to the PPPoE server.

The PPPoE over Ethernet feature is described in the following topics:

Feature History for PPPoE over Ethernet

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for PPPoE over Ethernet

The PPPoE over Ethernet feature has the following restriction:

- The Cisco 10000 series router currently supports the PPPoE over Ethernet feature on Gigabit Ethernet line cards and Fast Ethernet 8-port half-height line cards. The Fast Ethernet port of the performance routing engine (PRE) does not support the PPPoE over Ethernet feature.

Note The Cisco 10000 series router supports a Fast Ethernet interface for management traffic only.

Configuration Tasks for PPPoE over Ethernet

To configure the PPPoE over Ethernet feature, perform the following configuration tasks:

- Configuring a Virtual Template Interface, page 6-2
- Creating an Ethernet Interface and Enabling PPPoE, page 6-3
- Configuring PPPoE in a VPDN Group, page 6-3
- Configuring PPPoE in a BBA Group, page 6-3

Configuring a Virtual Template Interface

Configure a virtual template before you configure PPPoE on an Ethernet interface. The virtual template interface is a logical entity that is applied dynamically as needed to an incoming PPP session request. To configure a virtual template interface, see the "Configuring a Virtual Template Interface" section on page 3-17.

Creating an Ethernet Interface and Enabling PPPoE

To create an Ethernet interface and enable PPPoE on it, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface GigabitEthernet number`	Creates an Ethernet interface and enters interface configuration mode.
Step 2	`Router(config-if)# pppoe enable`	Enables PPPoE and allows PPPoE sessions to be created through that interface.

Configuring PPPoE in a VPDN Group

To configure a virtual private dial network (VPDN) group for PPPoE and to link the group to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN configuration on the router.
Step 2	`Router(config)# vpdn-group name`	Associates a VPDN group to a customer or VPDN profile.
Step 3	`Router(config-vpdn)# accept-dialin`	Creates an accept dial-in VPDN group.
Step 4	`Router(config-vpdn-acc-in)# protocol pppoe`	Specifies the VPDN group to be used to establish PPPoE sessions.
Step 5	`Router(config-vpdn-acc-in)# virtual-template template-number`	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 6	`Router(config-vpdn)# pppoe limit per-mac per-mac-limit`	(Optional) Specifies the maximum number of sessions per MAC address for each PPPoE port that uses the group.
Step 7	`Router(config-vpdn)# pppoe limit max-sessions number`	(Optional) Specifies the maximum number of PPPoE sessions that can be terminated on this router from all interfaces.

Note You cannot simultaneously configure a broadband aggregation (BBA) group for PPPoE and a VPDN group for PPPoE. If you configure a BBA group and then you configure a VPDN group, the **protocol** command in VPDN accept-dialin configuration mode does not include an option for PPPoE (for example, you cannot specify the **protocol pppoe** command). Use the **no bba-group pppoe** command to re-enable the **pppoe** option for the **protocol** command.

Configuring PPPoE in a BBA Group

Note Cisco IOS Release 12.2(15)BX does not support the configuration of BBA groups using RADIUS. You must configure BBA groups manually.

To configure a broadband aggregation (BBA) group for PPPoE and to link it to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# bba-group pppoe {name | global}`	Configures a BBA group to be used to establish PPPoE sessions. *name* identifies the BBA group. You can have multiple BBA groups. **global** is the default BBA group used for ATM connections when a BBA group name is not specified.
Step 2	`Router(config-bba)# virtual-template template-number`	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 3	`Router(config-bba)# pppoe limit per-mac per-mac-limit`	(Optional) Specifies the maximum number of sessions per MAC address for each PPPoE port that uses the group.
Step 4	`Router(config-bba)# pppoe limit max-sessions number`	(Optional) Specifies the maximum number of PPPoE sessions that can be terminated on this router from all interfaces.
Step 5	`Router(config-bba)# pppoe limit per-vc per-vc-limit`	(Optional) Specifies the maximum number of PPPoE sessions for each VC that uses the group.
Step 6	`Router(config-bba)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# interface type number`	Specifies the interface to which you want to attach the BBA group and enters interface configuration mode.
Step 8	`Router(config-if)# encapsulation dot1q vlan-id`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN. Specify the VLAN identifier.
Step 9	`Router(config-if)# protocol pppoe group group-name`	Attaches the BBA group to the VLAN.

Note You cannot simultaneously configure a BBA group for PPPoE and a VPDN group for PPPoE. If you configure a BBA group and then you configure a VPDN group, the **protocol** command in VPDN accept-dialin configuration mode does not include an option for PPPoE (for example, you cannot specify the **protocol pppoe** command). Use the **no bba-group pppoe** command to re-enable the **pppoe** option for the **protocol** command.

Configuration Example for PPPoE over Ethernet

Example 6-1 shows a PPPoE over Ethernet configuration. In the example, the virtual template *virtual-template 1* is linked to the VPDN group. The configuration also specifies the number of sessions allowed on the VPDN group.

Example 6-1 Using a VPDN Group to Configure PPPoE over Ethernet

```
!Creates a VPDN session group and links it to a virtual template.
vpdn-group 1
    accept-dialin
        protocol pppoe
        virtual-template 1
    pppoe limit per-mac 10
    pppoe limit max-sessions 32000

interface Loopback0
    ip address 172.16.0.1 255.255.255.255

!Enables PPPoE and allows PPPoE sessions to be created through this subinterface.
interface GigabitEthernet1/0/0
    no ip address
    negotiation auto
    pppoe enable

!Configures the virtual template interface.
interface Virtual-Template1
    ip unnumbered loop 0
    mtu 1492
    peer default ip address pool pool1
    ppp authentication chap

!Specifies the IP local pool to use for address assignment.
ip local pool pool1 192.168.0.1 192.168.0.100
```

Example 6-2 creates a BBA group named *vpn-1* and links it to virtual-template 1. The vpn-1 BBA group is associated with VLAN 20.

Example 6-2 Using a BBA Group to Configure PPPoE over Ethernet

```
bba-group pppoe vpn-1
    virtual-template 1
    sessions per-vc limit 5
    sessions per-mac limit 10
!
!
interface GigabitEthernet1/0/0.1
    encapsulation dot1q 20
        protocol pppoe group vpn-1
```

Static MAC Address for PPPoE

The Static MAC Address for PPPoE feature allows you to choose the MAC address to be used as the source MAC address for PPPoE over ATM sessions on ATM permanent virtual circuits (PVCs). You can configure this feature for either a broadband aggregation (BBA) group or a virtual private dialup network (VPDN) group. The feature is applied to all PPPoEoA sessions on ATM PVCs to which the BBA group or the VPDN group is applied.

Note Although the Static MAC Address for PPPoE feature is configurable for VPDN groups, we recommend that you configure this feature for BBA groups.

The configuration of the Static MAC Address for PPPoE feature for BBA groups and VPDN groups is mutually exclusive. If you configure a MAC address as a source MAC address for a BBA group, a VPDN group cannot use this MAC address as a source MAC address for the VPDN group. To apply the BBA group MAC address to a VPDN group, you must manually configure the Static MAC Address for PPPoE feature for the VPDN group as well.

Example 6-3 shows how you can throttle PPP sessions using the MAC address. This example allows a maximum of five sessions from each MAC address. If more than five sessions are attempted from this MAC address, any sessions using that particular MAC address are throttled for 30 seconds.

Example 6-3 Throttling PPP Sessions Using the MAC Address

```
bba-group pppoe PPPoE
 virtual-template 1
 sessions per-vc limit 32000
 sessions per-mac limit 32000
 sessions per-mac throttle 5 1 30
```

To get a list of the throttled MAC addresses, use the **show pppoe throttled mac** command in privileged EXEC mode:

```
Router# show pppoe throttled mac
 MAC(s) throttled
     MAC              Ingress Port
00c1.00aa.006c          ATM1/0/0.101
007c.009e.0070          ATM1/0/0.101
0097.009d.007a          ATM1/0/0.101
008c.0077.0082          ATM1/0/0.101
00b5.00a8.009f          ATM1/0/0.101
00a4.0088.00b5          ATM1/0/0.101
```

Feature History for Static MAC Address for PPPoE

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

PPPoE over IEEE 802.1Q VLANs

The PPPoE over IEEE 802.1Q VLANs feature enables the Cisco 10000 series router to support PPPoE over IEEE 802.1Q encapsulated VLAN interfaces. IEEE 802.1Q encapsulation is used to interconnect a VLAN-capable router with another VLAN-capable networking device. The packets on the 802.1Q link contain a standard Ethernet frame and the VLAN information associated with that frame.

Note PPPoE is disabled by default on a VLAN.

The PPPoE over IEEE 802.1Q VLANs feature is described in the following topics:

Feature History for PPPoE over IEEE 802.1Q VLANs

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for PPPoE over IEEE 802.1Q VLANs

The PPPoE over IEEE 802.1Q VLANs feature has the following restrictions:

- The Cisco 10000 series router currently supports the PPPoE over IEEE 802.1Q VLANs feature on Gigabit Ethernet line cards and Fast Ethernet 8-port half-height line cards. The Fast Ethernet port of the performance routing engine (PRE) does not support this feature.
- The Cisco 10000 series router supports this feature for PPPoE dialin only. PPPoE dialout (client) is not supported.

Configuration Tasks for PPPoE over IEEE 802.1Q VLANs

To configure the PPPoE over IEEE 802.1Q VLANs feature, perform the following configuration tasks:

- Configuring PPPoE in a VPDN Group, page 6-8
- Configuring PPPoE in a BBA Group, page 6-9

The following sections describe how to perform these configuration tasks. For more information, see the "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*.

Configuring a Virtual Template Interface

Configure a virtual template interface before you configure PPPoE on an IEEE 802.1Q VLAN interface. The virtual template interface is a logical entity that is applied dynamically as needed to a serial interface. To configure a virtual template interface, see the "Configuring a Virtual Template Interface" section on page 3-17.

Creating an Ethernet 802.1Q Encapsulated Subinterface and Enabling PPPoE

To create an Ethernet 802.1Q encapsulated subinterface and enable PPPoE on it, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface GigabitEthernet slot/module/port.subinterface-number`	Creates a Gigabit Ethernet subinterface and enters subinterface configuration mode.
Step 2	`Router(config-subif)# encapsulation dot1q vlan-id`	Enables IEEE 802.1Q encapsulation on a specified subinterface in VLANs.
Step 3	`Router(config-subif)# pppoe enable`	Enables PPPoE and allows PPPoE sessions to be created through the specified subinterface.
Step 4	`Router(config-subif)# pppoe max-sessions number`	Specifies the maximum number of PPPoE sessions that can be terminated on this router from all interfaces.

Configuring PPPoE in a VPDN Group

To configure a VPDN group for PPPoE and link it to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN configuration on the router.
Step 2	`Router(config)# vpdn-group name`	Associates a VPDN group to a customer or VPDN profile.

	Command	Purpose
Step 3	`Router(config-vpdn)# accept-dialin`	Creates an accept dial-in VPDN group.
Step 4	`Router(config-vpdn-acc-in)# protocol pppoe`	Specifies the VPDN group to be used to establish PPPoE sessions.
Step 5	`Router(config-vpdn-acc-in)# virtual-template` *template-number*	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 6	`Router(config-vpdn)# pppoe limit per-vlan` *number*	(Optional) Specifies the maximum number of PPPoE sessions under each VLAN.
Step 7	`Router(config-vpdn)# pppoe limit per-mac` *per-mac-limit*	(Optional) Specifies the maximum number of sessions per MAC address for each PPPoE port that uses the group.
Step 8	`Router(config-vpdn)# pppoe limit max-sessions` *number*	(Optional) Specifies the maximum number of PPPoE sessions that can be terminated on this router from all interfaces.

Note You cannot simultaneously configure a broadband aggregation (BBA) group for PPPoE and a VPDN group for PPPoE. If you configure a BBA group and then you configure a VPDN group, the **protocol** command in VPDN accept-dialin configuration mode does not include an option for PPPoE (for example, you cannot specify the **protocol pppoe** command). Use the **no bba-group pppoe** command to re-enable the **pppoe** option for the **protocol** command.

Configuring PPPoE in a BBA Group

Note Cisco IOS Release 12.2(15)BX does not support the configuration of BBA groups using RADIUS. You must configure BBA groups manually.

To configure a broadband aggregation (BBA) group for PPPoE and to link it to the appropriate virtual template interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# bba-group pppoe` {*name* | **global**}	Configures a BBA group to be used to establish PPPoE sessions. *name* identifies the BBA group. You can have multiple BBA groups. **global** is the default BBA group used for ATM connections when a BBA group name is not specified.
Step 2	`Router(config-bba)# virtual-template` *template-number*	Specifies the virtual template interface to use to clone virtual access interfaces (VAIs).
Step 3	`Router(config-bba)# pppoe limit per-mac` *per-mac-limit*	(Optional) Specifies the maximum number of sessions per MAC address for each PPPoE port that uses the group.
Step 4	`Router(config-bba)# pppoe limit per-vlan` *number*	(Optional) Specifies the maximum number of PPPoE sessions under each VLAN.
Step 5	`Router(config-bba)# pppoe limit max-sessions` *number*	(Optional) Specifies the maximum number of PPPoE sessions to be terminated on this router from all interfaces.

	Command	Purpose
Step 6	`Router(config-bba)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# interface type number`	Specifies the interface to which you want to attach the BBA group and enters interface configuration mode.
Step 8	`Router(config-if)# encapsulation dot1q vlan-id`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN. Specify the VLAN identifier.
Step 9	`Router(config-if)# protocol pppoe group group-name`	Attaches the BBA group to the VLAN.

Note You cannot simultaneously configure a BBA group for PPPoE and a VPDN group for PPPoE. If you configure a BBA group and then you configure a VPDN group, the **protocol** command in VPDN accept-dialin configuration mode does not include an option for PPPoE (for example, you cannot specify the **protocol pppoe** command). Use the **no bba-group pppoe** command to re-enable the **pppoe** option for the **protocol** command.

Configuration Examples for PPPoE over IEEE 802.1Q VLANs

Example 6-4 shows a PPPoE over IEEE 802.1Q encapsulated VLAN configuration. In the example, the virtual-template 1 virtual template is linked to the VPDN group. The configuration also specifies the maximum number of sessions allowed on the VPDN group and the number of sessions allowed for each VLAN.

Example 6-4 Using a VPDN Group to Configure PPPoE over IEEE 802.1Q VLANs

```
!Enables a virtual private dial-up network configuration on the router.
vpdn enable
!
!Creates a VPDN session group and links it to a virtual template.
vpdn-group 1
    accept-dialin
        protocol pppoe
        virtual-template 1
    pppoe limit per-mac 10
    pppoe limit per-vlan 100
    pppoe limit max-sessions 32000

interface Loopback0
    ip address 172.16.0.1 255.255.255.255

interface GigabitEthernet1/0/0
    no ip address
    negotiation auto

!Enables PPPoE and allows PPPoE sessions to be created through this subinterface.
interface GigabitEthernet1/0/0.10
    encapsulation dot1Q 20
    pppoe enable
    pppoe max-sessions 10

!Configures the virtual template interface.
interface Virtual-Template1
    ip unnumbered loop 0
    mtu 1492
```

```
    peer default ip address pool pool1
    ppp authentication chap

!Specifies the IP local pool to use for address assignment.
ip local pool pool1 192.168.0.1 192.168.0.100
```

Example 6-5 creates two BBA groups: *VPN_1* and *VPN_2*. The VPN_1 BBA group is associated with *virtual-template 1* and the VPN_2 BBA group is associated with *virtual-template 2*. The VPN_1 group is associated with VLAN 20 and the VPN_2 group is associated with VLAN 30.

Example 6-5 Using a BBA Group to Configure PPPoE over IEEE 802.1Q VLANs

```
bba-group pppoe VPN_1
    virtual-template 1
    sessions per-vc limit 5
    sessions per-mac limit 10
    sessions per-vlan limit 5
!
!
bba-group pppoe VPN_2
    virtual-template 2
    sessions per-vc limit 5
    sessions per-mac limit 10
    sessions per-vlan limit 5
!
!
interface GigabitEthernet1/0/0.1
    encapsulation dot1q 20
        protocol pppoe group VPN_1
!
interface GigabitEthernet 2/0/0.2
    encapsulation dot1q 30
        protocol pppoe group VPN_2
```

Verifying PPPoE over Ethernet and IEEE 802.1Q VLAN

To verify PPPoE over Ethernet and IEEE 802.1Q VLAN, enter the following commands in privileged EXEC mode:

Command	Purpose
Router# **show vpdn**	Displays information about active Level 2 Forwarding (L2F) Protocol tunnel and message identifiers in a VPDN.
Router# **show vpdn session**	Displays information about active Layer 2 Tunnel Protocol (L2TP) or Layer 2 Forwarding (L2F) sessions in a VPDN.
Router# **show vpdn session packet**	Displays PPPoE session statistics.
Router# **show vpdn session all**	Displays PPPoE session information for each session ID.
Router# **show vpdn tunnel**	Displays PPPoE session count for the tunnel.
Router# **show pppoe session all**	Displays PPPoE session information for each session ID.
Router# **show pppoe session packets**	Displays PPPoE session statistics.

Clearing PPPoE Sessions

To clear PPPoE sessions, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# clear pppoe all`	Clears all PPPoE sessions.
`Router# clear pppoe interface`	Clears all PPPoE sessions on a physical interface or subinterface.
`Router# clear pppoe rmac`	Clears PPPoE sessions from a client host MAC address.

TCP MSS Adjust

The TCP MSS Adjustment feature enables the configuration of the maximum packet segment size (MSS).

When a host (usually a PC) initiates a TCP session with a server, it negotiates the IP segment size by using the MSS option field in the TCP SYN packet. The value of the MSS field is determined by the maximum transmission unit (MTU) configuration on the host. The default MSS value for a PC is 1500 bytes.

The PPP over Ethernet (PPPoE) standard supports a MTU of only 1492 bytes. The disparity between the host and PPPoE MTU size can cause the router in between the host and the server to drop 1500-byte packets and terminate TCP sessions over the PPPoE network. Even if the path MTU (which detects the correct MTU across the path) is enabled on the host, sessions may be dropped because system administrators sometimes disable the ICMP error messages that must be relayed from the host in order for path MTU to work.

In most cases, the optimum value for the max-segment-size argument is 1452 bytes. This value plus the 20-byte IP header, the 20-byte TCP header, and the 8-byte PPPoE header add up to a 1500-byte packet that matches the MTU size for the Ethernet link.

Feature History for TCP MSS Adjust

Cisco IOS Release	Description	Required PRE
12.2(31)SB3	This feature was introduced on the Cisco 10000 series router.	PRE2 or PRE3

Information about TCP MSS Adjust

- This feature works for both PTA and LNS sessions.
- The MSS value is configured globally, so every packet transiting through the router are subject to a rewrite.
- The **per interface** command is only applicable to packets that get punted to the RP, so it is not recommended to use this command.

Restrictions for TCP MSS Adjust

- The TCP MSS Adjust feature only works if the **MaxSegSize** option is the first option included in the packet. If a non-typical TCP packet is received, where MaxSegSize is not the first option in the packet, the TCP MSS Adjust feature configuration will have no effect.

Configuration Task for TCP MSS Adjust

Perform this task to configure the maximum segment size (MSS) for transient packets that traverse the Cisco 10000 Series router, specifically TCP segments in the SYN bit and to configure the MTU size of IP packets.

SUMMARY STEPS

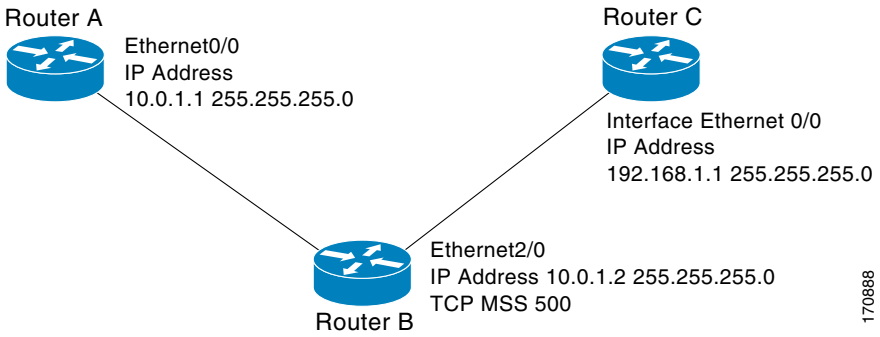
1. **enable**
2. **configure terminal**
3. **ip pxf adjust-mss** *max-segment-size*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip pxf adjust-mss** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1452`	Adjusts the MSS value of TCP SYN packets going through the Cisco 10000 Series router. The *max-segment-size* argument is the maximum segment size, in bytes. The range is from 500 to 1460.
Step 4	**end** **Example:** `Router(config-if)# end`	Exits to global configuration mode.

TCP MSS Adjustment Configuration: Examples

Figure 1 Example Topology for TCP MSS Adjustment



The following example shows how to configure and verify the adjustment value. Configure the interface adjustment value on router B:

```
Router_B(config)# ip pxf adjust-mss 500
```

Telnet from router A to router C, with B having the MSS adjustment configured.

```
Router_A# telnet 192.168.1.1
Trying 192.168.1.1... Open
```

Observe the debug output from router C:

```
Router_C# debug ip tcp transactions

Sep 5 18:42:46.247: TCP0: state was LISTEN -> SYNRCVD [23 -> 10.0.1.1(38437)]
Sep 5 18:42:46.247: TCP: tcb 32290C0 connection to 10.0.1.1:38437, peer MSS 500, MSS is
500
Sep 5 18:42:46.247: TCP: sending SYN, seq 580539401, ack 6015751
Sep 5 18:42:46.247: TCP0: Connection to 10.0.1.1:38437, advertising MSS 500
Sep 5 18:42:46.251: TCP0: state was SYNRCVD -> ESTAB [23 -> 10.0.1.1(38437)]
```

The MSS gets adjusted to 500 on Router_B as configured.

The following example shows the configuration of a PPPoE client with the MSS value set to 1452:

```
vpdn enable
no vpdn logging
!
vpdn-group 1
request-dialin
protocol pppoe
!
interface Ethernet0
 ip address 192.168.100.1.255.255.255.0
 ip pxf adjust-mss 1452
 ip nat inside
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 8/35
 pppoe client dial-pool-number 1
!
dsl equipment-type CPE
dsl operating-mode GSHDSL symmetric annex B
dsl linerate AUTO
```

```
!
interface Dialer1
 ip address negotiated
 ip mtu 1492
 ip nat outside
 encapsulation ppp
 dialer pool 1
 dialer-group 1
 ppp authentication pap callin
 ppp pap sent-username sohodyn password 7 141B1309000528
!
ip nat inside source list 101 Dialer1 overload
ip route 0.0.0.0.0.0.0.0 Dialer1
access-list permit ip 192.168.100.0.0.0.0.255 any
```

VLAN Range

The VLAN range feature simplifies the configuration of VLAN subinterfaces. By using this feature, you can configure a group of VLAN subinterfaces at one time instead of configuring each subinterface separately. The commands you enter for a group of VLAN subinterfaces apply to each subinterface within the group and are applied to all existing VLANs.

By using the VLAN range feature, you can also configure overlapping ranges of subinterfaces and an individual subinterface within a range of subinterfaces.

The VLAN Range feature is described in the following topics:

Feature History for VLAN Range

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for VLAN Range

The VLAN range feature has the following restrictions:

- The commands you enter in interface range configuration mode (the mode you enter after issuing the **interface range** command) are executed as you enter them. The commands are not batched together for execution after you exit interface range mode. If you exit interface range configuration mode while the commands are being executed, some commands might not be executed on some interfaces in the range. Wait until the command prompt reappears before exiting interface range configuration mode.
- All configuration changes made to a range of subinterfaces are saved to NVRAM, but the range itself does not get saved to NVRAM. To create and save a range, enter the **define interface-range** global configuration command.
- Cisco IOS software does not support the **no interface range** command. To delete a range of subinterfaces, you must delete the individual subinterfaces.

Configuration Task for VLAN Range

To configure the VLAN range feature, perform the following required configuration task:

- Configuring a Range of VLAN Subinterfaces, page 6-16

Configuring a Range of VLAN Subinterfaces

To configure a range of VLAN subinterfaces, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface range {{ethernet | fastethernet | gigabitethernet | atm} slot/interface.subinterface - {ethernet | fastethernet | gigabitethernet | atm} slot/interface.subinterface}`	Selects the range of subinterfaces to be configured. If you specify subinterfaces that have not been previously created, the **interface range** command creates the subinterfaces. Enters interface range configuration mode. **Note** The spaces around the dash are required. For example, the command **interface range fastethernet 1 - 5** is valid; the command **interface range fastethernet 1-5** is not valid.

	Command	Purpose
Step 2	`Router(config-int-range)# encapsulation dot1q vlan-id [native]`	Enables IEEE 802.1Q encapsulation of traffic and applies a unique VLAN ID to each subinterface within the range. The *vlan-id* argument is the virtual LAN identifier. You must enter a value from 1 to 4095. **Note** VLAN ID 0 is a valid ID, but is not a valid designation of a VLAN. VLAN ID 0 is used primarily to convey class of service (CoS) information on packets that would otherwise be untagged. (Optional) The **native** argument sets the VLAN ID value of the port to the *vlan-id* value. **Note** The VLAN ID is applied to the first subinterface in the range. Each subsequent interface is assigned a VLAN ID, which is the specified *vlan-id* plus the subinterface number, minus the first subinterface number: `VLAN ID + subinterface number - first subinterface number`

Configuration Examples for VLAN Range

Example 6-6 configures the Fast Ethernet subinterfaces with the range 5/1.1 to 5/1.4 and applies the following VLAN IDs to the subinterfaces:

- Fast Ethernet5/1.1 = VLAN ID 301 (*vlan-id*)
- Fast Ethernet5/1.2 = VLAN ID 302 (*vlan-id* = 301 + 2 - 1 = 302)
- Fast Ethernet5/1.3 = VLAN ID 303 (*vlan-id* = 301 + 3 - 1 = 303)
- Fast Ethernet5/1.4 = VLAN ID 304 (*vlan-id* = 301 + 4 - 1 = 304)

Example 6-6 Configuring a Range of VLAN Subinterfaces

```
Router(config)# interface range fastethernet5/1.1 - fastethernet5/1.4
Router(config-if-range)# encapsulation dot1q 301
Router(config-if-range)# no shutdown
```

Verifying the Configuration of a Range of Subinterfaces

To verify the configuration of a range of subinterfaces for VLAN encapsulation, enter the following commands in privilege EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the current configuration, including information about the interfaces and subinterfaces configured on the router and the type of encapsulation configured for each interface.
`Router# show interface`	Displays information about all interfaces and subinterfaces configured on the router, including the type of encapsulation configured for each interface.
`Router# show interface` *interface-type slot*`/`*interface.subinterface*	Displays information about the interface or subinterface you specify, including the type of encapsulation configured.

CHAPTER 7

Configuring IP Unnumbered on IEEE 802.1Q VLANs

Service providers continuously seek ways in which they can make their networks less complex and less expensive, and reduce the cost of provisioning subscribers. One way in which service providers can achieve these results is to migrate their ATM networks to IP networks and upgrade their DSLAM to use a Gigabit Ethernet uplink, instead of an ATM uplink, to connect their DSLAM to an aggregation router, such as the Cisco 10000 series router.

In the Digital Subscriber Line (DSL) environment, service providers use a service model that configures ATM Routed Bridge Encapsulation (RBE) on an unnumbered interface of the aggregation router. This configuration associates an IP route with a subscriber, and uses a virtual path identifier/virtual connection identifier (VPI/VCI) pair to identify the IP route. In this way, all subscribers can securely share the same subnet, which enables the service provider to save IP address space. When the DHCP server provides an IP address to the subscriber, the aggregation router dynamically configures the IP route.

The Cisco 10000 series router builds on the RBE on an unnumbered interface service model to enable you to configure IP unnumbered on IEEE 802.1Q VLANs. Instead of using a VPI/VCI pair to identify a subscriber route, the Cisco 10000 series router maps a VLAN identifier to the subscriber on an Ethernet interface.

The Cisco 10000 series router supports the IP Unnumbered on IEEE 802.1Q VLANs feature. Prior to Cisco IOS Release 12.3(7)XI1, IP support for VLAN subinterfaces required that you configure separate IP subnets for each of the subinterfaces that terminate the VLAN. This resulted in inefficient use of the IP address space because an entire IP subnet is often not needed for the hosts assigned to a VLAN. The IP Unnumbered on VLANs feature helps to conserve IP address space for service provider configurations that include Ethernet VLAN subinterfaces.

VLAN subinterfaces with IP unnumbered configured support DHCP for IP address allocation. The DHCP server uses the information in DHCP Option 82 to assign IP addresses to the hosts on a VLAN. The routing table is dynamically updated to insert an IP route for the IP address assigned on each of the subinterfaces. These IP host routes exist until the DHCP lease time expires or the host releases the leased address.

Note For more information about Option 82, see the "DHCP Relay Agent Information Option—Option 82" section on page 3-9.

When a subinterface goes down, the IP host route exists until the DHCP lease time expires. However, if you enter the **show ip route dhcp** command, the IP host routes do not display. After the subinterface comes back up, the IP host routes display when you enter the **show ip route dhcp** command if the DHCP lease time has not expired.

This chapter describes the IP Unnumbered on IEEE 802.1Q VLANs feature in the following topics:

Feature History for IP Unnumbered on VLANs

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Benefits for IP Unnumbered on VLANs

The IP Unnumbered on VLANs feature benefits service providers in the following ways:

- DSL providers can easily migrate their ATM networks to IP networks and migrate their DSLAMs from an ATM uplink to a Gigabit Ethernet uplink for connection to the router.
- Using one router and the same service model, providers can aggregate Layer 2 access for DSL and Metro-Ethernet subscribers.
- IP address space is saved because all ports can share the same subnet.
- Each IP unnumbered subinterface supports one VLAN and each VLAN can have multiple IP addresses.
- Subscribers are easily identified, which makes it possible to apply different policies on a per-subscriber basis.
- DHCP simplifies address management.
- The use of VLANs increases security.
- Security is greater with the use of VLANs. Because routing information is obtained from DHCP, ARP and MAC entries cannot be spoofed.

Restrictions for IP Unnumbered on VLANs

The IP Unnumbered on VLANs feature has the following restrictions:

- You can configure IP unnumbered on only Ethernet VLAN subinterfaces and point-to-point interfaces.
- If you configure more than 14,000 IP unnumbered subinterfaces and you have configured EIGRP on all interfaces on a router, the router can stop responding. To avoid this problem, use the **passive-interface default** command (which disables all router interfaces from sending routing updates) and then configure the **no passive-interface** command on selected interfaces you want to send routing updates.

Configuration Tasks for IP Unnumbered on VLANs

To configure the IP Unnumbered on VLANs feature, perform at least one of the following configuration tasks:

- Configuring IP Unnumbered for an Ethernet VLAN Subinterface, page 7-3
- Configuring IP Unnumbered for a Range of Ethernet VLAN Subinterfaces, page 7-4

Configuring IP Unnumbered for an Ethernet VLAN Subinterface

To configure IP unnumbered for an Ethernet VLAN subinterface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number slot/module/port.subinterface`	Configures a subinterface and enters subinterface configuration mode.
Step 2	`Router(config-subif)# encapsulation dot1q vlan-id [native]`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a virtual LAN (VLAN). IEEE 802.1 Q enscapsulation is disabled by default. The *vlan-id* argument is the virtual LAN identifier. Valid values are from 1 to 4095. The **native** option sets the VLAN ID value of the port to the value you specify in the *vlan-id* argument.
Step 3	`Router(config-subif)# ip unnumbered type number`	Enables IP processing on a serial interface without assigning an explicit IP address to the interface. IP unnumbered is disabled by default. The *type* and *number* arguments indicate the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.

Example 7-1 configures IP unnumbered on the Fast Ethernet 1/0.1 subinterface.

Example 7-1 Configuring IP Unnumbered on an Ethernet VLAN Subinterface

```
Router(config)# interface fastethernet 1/0.1
Router(config-subif)# encapsulation dot1q 10
Router(config-subif)# ip unnumbered ethernet3/0
```

Configuring IP Unnumbered for a Range of Ethernet VLAN Subinterfaces

To configure IP unnumber on a range of Ethernet VLAN subinterfaces, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface range` *type number slot/module/port.subinterface* `-` *type number slot/module/port.subinterface*	Configures a range of subinterfaces and enters subinterface-range configuration mode.
Step 2	`Router(config-subif-range)# encapsulation dot1q` *vlan-id* `[native]`	Applies a VLAN ID to each subinterface within the range you specify using the **interface range** command. The VLAN ID that you specify in the *vlan-id* argument is applied to the first subinterface in the range. Each subsequent interface is assigned a VLAN ID, which is the specified *vlan-id* plus the subinterface number minus the first subinterface number. For example: VLAN ID + subinterface number - first subinterface number
Step 3	`Router(config-subif-range)# ip unnumbered` *type number*	Enables IP processing on a serial interface without assigning an explicit IP address to the interface. IP unnumbered is disabled by default. The *type* and *number* arguments indicate the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.

Example 7-2 configures IP unnumbered on the Fast Ethernet subinterfaces 1/0.1 to 1/0.1000.

Example 7-2 Configuring IP Unnumbered on a Range of Ethernet VLAN Subinterfaces

```
Router(config)# interface range fastethernet 1/0.1 - fastethernet 1/0.1000
Router(config-subif-range)# ip unnumbered ethernet 3/0
```

Configuration Examples for IP Unnumbered on VLANs

The following example enables IP unnumbered on the Fast Ethernet 0/0.1 VLAN subinterface:

```
!
interface fastethernet0/0.1
 encapsulation dot1q 101
 ip unnumbered ethernet 0
```

The following example enables IP unnumbered on a range of VLAN subinterfaces:

```
interface range fastethernet0/0.11 - fastethernet0/0.60
 encapsulation dot1q 101
 ip unnumbered ethernet 0
```

Monitoring and Maintaining IP Unnumbered Ethernet VLAN Subinterfaces

To monitor and maintain IP unnumbered Ethernet VLAN subinterfaces, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show interfaces** *type number slot/module/port.subinterface*	Displays information about the interface you specify.
Router# **show running-config**	Displays the contents of the currently running configuration file.
Router# **show running-config** [**interface** *type number*]	Displays the configuration for a specific interface.
Router# **show vlans**	Displays information about VLAN subinterfaces.

CHAPTER 8

Configuring ATM Permanent Virtual Circuit Autoprovisioning

With the rapid growth in broadband customers, service providers need to provision service for subscribers in the most efficient and accurate way possible. The ATM PVC autoprovisioning feature automates the configuration of a large number of ATM permanent virtual circuits (PVCs) in DSL service provider networks using the PPPoA, PPPoE, and RBE protocols.

This chapter describes how to configure the ATM PVC Autoprovisioning feature by using a local configuration. It also describes the VBR-nrt Oversubscription feature for ATM VCs.

This chapter describes the following features

ATM PVC Autoprovisioning

The Cisco 10000 series router supports the ATM PVC Autoprovisioning feature. By using this feature, DSL wholesale service providers can use a local configuration to dynamically provision ATM service for subscribers.

Incoming traffic on the VPI/VCI pair triggers virtual circuit (VC) creation. The Cisco 10000 series router does not create the on-demand VC until incoming traffic arrives. For example:

- On-demand VCs configured on the interface remain in the inactive state until the first incoming packet arrives on the VC, triggering VC creation.
- If you use the **reload** command on the Cisco 10000 series router, the router does not establish the on-demand VCs until incoming traffic triggers VC creation.

The ATM PVC Autoprovisioning feature is described in the following topics:

Local Template-Based ATM PVC Provisioning

The Local Template-Based ATM PVC Provisioning feature supports PVC autoprovisioning for an infinite range of VPI/VCI combinations on an ATM interface.

The Local Template-Based ATM PVC Provisioning feature enables ATM permanent virtual circuits (PVCs) to be provisioned automatically as needed from a local configuration, making the provisioning of large numbers of digital subscriber line (DSL) subscribers easier, faster, and less prone to error. ATM PVC autoprovisioning can be configured on a PVC, an ATM PVC range, or a VC class. If a VC class configured with ATM PVC autoprovisioning is assigned to the *main* interface, all the PVCs on that main interface will be autoprovisioned; this configuration is sometimes referred to as an infinite range.

In releases earlier than Cisco IOS Release 12.3(7)XI2, a reassembly channel had to be opened to receive any incoming packets on a create-on-demand VC. In Cisco IOS Release 12.3(7)XI2, the SAR sends the cell header to the RP, thus removing the need to open the reassembly channel to receive cells. Any cells received on an unopened channel result in cell headers from the SAR until the VC is opened.

Autoprovisioned ATM PVCs are not created until there is activity on the virtual path identifier (VPI)/virtual channel identifier (VCI) pair. When the interface is disabled and re-enabled using the **shutdown** and **no shutdown** commands, autoprovisioned PVCs that are part of a PVC range or infinite range are removed upon shutdown and are not reestablished until the first incoming cell triggers PVC creation. During router reload, autoprovisioned PVCs are created when there is activity on the connection.

Feature History for Local Template-Based ATM PVC Provisioning

Cisco IOS Release	Description	Required PRE
12.3(7)XI2	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

ATM Interface Oversubscription

The Cisco 10000 series router allows you to create more on-demand PVCs than the chassis allows to be active at the same time. For example, the router chassis allows a total of 61,500 PVCs to be up at the same time, even though you can configure more than 61,500 on-demand PVCs on the chassis. In actuality, you can configure up to 32,000 PVCs on each line card. If you install ATM line cards in six of the eight available slots in the chassis, you can configure up to 128,000 on-demand PVCs instead of the 61,500 PVCs chassis limit.

When the number of ATM PVCs exceeds a port's active VC count, you can use the **idle-timeout** interface command to dynamically bring down on-demand PVCs. If you use CLI commands to explicitly configure a PVC, the router brings the PVC to the inactive state when the idle-timeout timer expires.

VC Class

A VC class is a set of preconfigured VC parameters that you configure and apply to a particular VC or ATM interface. The VC parameters that you can configure for a VC class include the following:

- Attach ATM VC class to an interface (**class-int** command)
- Constant bit rate (**cbr** command)
- Encapsulation type (**encapsulation aal5** command)
- Idle-timeout (**idle-timeout** command)
- Integrated Local Management Interface (ILMI) management (**ilmi manage** command)
- Inverse ARP broadcasts (**protocol** command)
- Inverse ARP time period (**inarp** command)
- OAM management on a PVC (**oam-pvc** command)
- OAM management parameters for re-establishing and removing a PVC connection (**oam retry** command)
- PVC autoprovisioning (**create on-demand** command)
- Queue depth (**queue-depth** command)
- Static map for an ATM PVC or VC (**protocol** command)
- Unspecified bit rate (**ubr** command)
- Variable Bit Rate-Non Real Time quality of service (**vbr-nrt** command)
- Weight (**weight** command)

For more information, see the Configuring ATM chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*.

ATM VC Scaling and VC Assignment

The ATM line cards support the full range of VPI//VCI pairs (unidirection only)—8 bit VPI range and 16 bit VCI range. Table 8-1 lists the maximum number of active VCs supported on ATM line cards for Cisco IOS Release 12.3(7)XI2 or later releases.

Table 8-1 Active VCs on ATM Line Cards

Line Card	Max. VCs per Port	Maximum VCs per Module	No. VBR, CBR, Shaped UBR VCs
E3/DS3	4,096	32,768[1]	28,672[2]
OC-3	8,191	32,764[3]	28,672[4]
OC-12	16,384 (previously 14,436)	16,384	16,384

1. For 32,768 VCs per module, 4096 of them must be unshaped UBR VCs.
2. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.
3. For 32,764 VCs per module, 4096 of them must be unshaped UBR VCs.
4. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.

You can configure the maximum number of VCs across the ports in any fashion, provided that you do not exceed the per-port maximum.

Although the maximum number of VBR, CBR, and shaped UBR VCs per E3/DS3 and OC-3 ATM line card is 28,672 VCs, the router supports a maximum of 22,204 VBR, CBR, and shaped UBR VCs per line card that you can place within virtual path (VP) tunnels. If you attempt to bring up more than 22,204 VCs in a configuration that includes VP tunnels and VCs (hierarchical traffic shaping configuration), the VCs might not assign traffic correctly or the VCs might not come up at all. Be sure to limit the number of configured VBR, CBR, and shaped UBR VCs on an ATM card to less than 22,204 VCs if you place the VCs in VP tunnels.

For the OC-12 ATM line card, the router supports 16,384 VCs in VP tunnels.

A pair of unidirectional Segmentation and Reassembly (SAR) chips are used on the line cards, one for the transmit direction and the other for the receive direction. The architecture of the SAR places limits on the following values supported by the router's ATM interfaces:

- Maximum number of active VCs
- Maximum number of VPI combinations that can be configured
- Maximum number of VCI combinations that can be configured

To allow the SAR to support the same VPI/VCI values per interface and thus discriminate among the VCs, the SAR translates the external VPI/VCI values into an internal 32-bit logical header that includes bits for the port number. Router interfaces can support 510 (page 0 is unused due to a hardware limitation; page 511 is reserved for tunnels) unique combinations of the following bit fields:

- PHY bits to designate the physical interface of the PVC. The OC-3 line card requires 2 bits for the port number; the OC-12 line card doesn't require any bits; and the DS-2 line card requires 3 bits.
- 8 VPI bits (represents the entire VPI value)
- Upper 9 bits of VCI value (bits 7-15 of the VCI field)

For more information, see the *Understanding the Maximum Number of Active Virtual Circuits on Cisco ATM Router Interfaces* tech note.

When SAR the Page Limit is Reached

In releases earlier than Cisco IOS Release 12.3(7)XI2, if the SAR page limit was reached while you were creating ATM PVCs, the router continued to create ATM PVCs but they were inactive.

In Cisco IOS Release 12.3(7)XI2, the router checks the SAR page limit before creating an ATM PVC. If the SAR page limit has been reached, a message displays indicating that there are no more SAR pages available for the PVC. A message similar to the following message is displayed for both individual ATM PVCs and ATM PVCs that are configured within a range.

```
*Oct 28 21:09:26.535: SAR exhausted the number of pages available to create this VC
*Oct 28 21:09:26.535: %ATM-3-FAILCREATEVC: ATM failed to create VC(VCD=1173, VPI=147,
VCI=1408) on Interface ATM5/0/3, (Cause of the failure: VPI/VCI out of range.)
```

OC-12 ATM Line Card and VC Scaling

The SAR on the OC-12 ATM line card uses four priority levels, 0 through 3. Unspecified bit rate (UBR) and virtual protocol (VP) tunnel use priority 3; variable bit rate (VBR) uses priority 2; and constant bit rate (CBR) uses priority 0. Each priority level supports a maximum of 16,000 VCs.

Feature History for ATM PVC Autoprovisioning

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was integrated into Cisco IOS Release 12.2(15)BX.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for ATM PVC Autoprovisioning

The ATM PVC autoprovisioning feature has the following restriction:

- The Segmentation and Reassembly (SAR) chip on the OC-3 and OC-12 ATM line cards is responsible for all physical ports on the line card. Restrictions on how VCs are assigned might reduce the VC counts.
- The ATM line cards use a pair of unidirectional SAR chips to segment and reassemble cells based on priority levels. The architecture of the SAR limits the way in which you can assign VCs. In some configurations, the limitation of the SAR can reduce the VC counts from the maximum number typically support (for example, a maximum of 8000 VCs per port for the OC-3 and 16,000 per port for the OC-12). For more information, see the "ATM VC Scaling and VC Assignment" section on page 8-4.
- The SAR translates the external VPI/VCI values into an internal 32-bit logical header. Router interfaces can support *510* unique bit field combinations in the 32-bit logical header. While there are 512 total SAR pages, page 0 is unused due to a hardware limitation and page 511 is reserved for tunnels.

Note Note: The limit of 510 usable SAR pages in Cisco IOS Release 12.3(7)XI2 represents a reduction from the limit of 512 usable SAR pages in earlier releases.

- The Local Template-Based ATM PVC Provisioning feature (infinite range) can be configured only on a *main* ATM interface; that is, it cannot be configured on a subinterface. When you use the **class-int** command to attach an ATM VC class to a subinterface, the **create on-demand** command is ignored.
- PVCs or PVCs within a range specified as create on demand PVCs, count against the interface limit for configured PVCs, regardless of whether the PVCs become active. These PVCs count against the maximum number of VCs allowed per interface port.

Configuration Tasks for ATM PVC Autoprovisioning

To configure the ATM PVC autoprovisioning feature, perform one of the following tasks:

- Creating an On-Demand PVC Using a VC Class, page 8-6
- Creating an On-Demand PVC Directly, page 8-8
- Creating an On-Demand PVC With Infinite Range, page 8-11

Creating an On-Demand PVC Using a VC Class

To create an on-demand PVC using a VC class, perform the following tasks:

- Creating a VC Class with PVC Autoprovisioning Enabled, page 8-6
- Applying the VC Class, page 8-7

Creating a VC Class with PVC Autoprovisioning Enabled

To create a VC class with the ATM PVC autoprovisioning feature enabled, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vc-class atm name`	Creates a VC class and enters vc-class configuration mode.
Step 2	`Router(config-vc-class)# create on-demand`	Enables PVC autoprovisioning. **Note** Configure additional VC parameters as appropriate. For more information, see the "VC Class" section on page 8-3 or the "Creating a VC Class Examples" section in the *Cisco IOS Wide-Area Networking Configuration Guide*.
Step 3	`Router(config-vc-class)# idle-timeout [time-out-in-seconds] [minimum-traffic-in-kbps]`	(Optional) Enables the idle-timeout timer on the on-demand PVC. The default *time-out-in-seconds* is 0 (no idle-timeout). The Cisco 10000 series router waits until the traffic on a particular VC is processed before tearing down the VC, even if you specify the *minimum-traffic-in-kbps* option or if the VC is idle during the idle-timeout period.

Example 8-1 creates a VC class named *myclass*, enables PVC autoprovisioning on the class, and sets the idle-timeout timer for 300 seconds. The configuration of the idle-timeout timer is optional.

Example 8-1 Configuring a VC Class with PVC Autoprovisioning Enabled

```
Router(config)# vc-class atm myclass
Router(config-vc-class)# create on-demand
Router(config-vc-class)# idle-timeout 300
```

Applying the VC Class

To apply a VC class, perform the following tasks:

- Applying a VC Class to an Individual PVC, page 8-7
- Applying a VC Class to a Range of PVCs, page 8-7
- Applying a VC Class to a Specific PVC Within a PVC Range, page 8-8

Applying a VC Class to an Individual PVC

To apply a VC class to a specific PVC, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.
Step 2	`Router(config-if)# pvc [name] vpi/vci`	Specifies the ATM PVC and enters atm-vc configuration mode.
Step 3	`Router(config-if-atm-vc)# class-vc vc-class-name`	Applies the VC class on the PVC.

Example 8-2 applies the VC class *myclass* to PVC 100/100.

Example 8-2 Applying a VC Class to an Individual PVC

```
Router(config)# interface atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# pvc 100/100
Router(config-subif-atm-vc)# class-vc myclass
```

Applying a VC Class to a Range of PVCs

To apply a VC class to a range of PVCs, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.

	Command	Purpose
Step 2	`Router(config-if)# range [range-name] pvc start-vpi/start-vci end-vpi/end-vci`	Specifies the range of PVCs and enters atm-range configuration mode.
Step 3	`Router(config-if-atm-range)# class-range class-name`	Applies the VC class on the range of PVCs.

Example 8-3 applies the VC class *myclass* to the PVC range 100/100 to 100/200.

Example 8-3 Applying a VC Class to a PVC Range

```
Router(config)# int atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# range pvc 100/100 100/200
Router(config-subif-atm-range)# class-range myclass
```

Applying a VC Class to a Specific PVC Within a PVC Range

To apply a VC class to a specific PVC within a PVC range, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.
Step 2	`Router(config-if)# range [range-name] pvc start-vpi/start-vci end-vpi/end-vci`	Specifies the range of PVCs and enters atm-range configuration mode.
Step 3	`Router(config-if-atm-range)# pvc-in-range [pvc-name] [vpi/vci]`	Specifies an individual PVC within a PVC range.
Step 4	`Router(config-if-atm-range-pvc)# class-vc vc-class-name`	Applies the VC class on the individual PVC within the PVC range.

Example 8-4 applies the VC class *myclass* to PVC 100/100 in the PVC range 100/100 to 100/200.

Example 8-4 Applying a VC Class to a Specific PVC Within a PVC Range

```
Router(config)# int atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# range pvc 100/100 100/200
Router(config-subif-atm-range)# pvc-in-range 100/100
Router(config-subif-atm-range-pvc)# class-vc myclass
```

Creating an On-Demand PVC Directly

To configure an on-demand PVC directly on an individual PVC, a PVC range, or a specific PVC within PVC range, perform the following tasks:

- Enabling ATM PVC Autoprovisioning on an Individual PVC, page 8-9
- Enabling ATM PVC Autoprovisioning on a Range of PVCs, page 8-9
- Enabling ATM PVC Autoprovisioning on a Specific PVC Within a PVC Range, page 8-10

Enabling ATM PVC Autoprovisioning on an Individual PVC

To enable ATM PVC autoprovisioning on an individual PVC, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.
Step 2	`Router(config-if)# pvc [name] vpi/vci`	Specifies the ATM PVC and enters atm-vc configuration mode.
Step 3	`Router(config-if-atm-vc)# create on-demand`	Enables PVC autoprovisioning on the individual PVC.
Step 4	`Router(config-if-atm-vc)# idle-timeout [time-out-in-seconds] [minimum-traffic-in-kbps]`	(Optional) Enables the idle-timeout timer on the individual on-demand PVC. The default *time-out-in-seconds* is 0 (no idle-timeout). The Cisco 10000 series router waits until the traffic on a particular VC is processed before tearing down the VC, even if you specify the *minimum-traffic-in-kbps* option or if the VC is idle during the idle-timeout period.

Example 8-5 enables autoprovisioning on PVC 100/100 and sets the idle-timeout timer for 300 seconds.

Example 8-5 Enabling ATM PVC Autoprovisioning on a PVC

```
Router(config)# int atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# pvc 100/100
Router(config-subif-atm-vc)# create on-demand
Router(config-subif-atm-vc)# idle-timeout 300
```

Enabling ATM PVC Autoprovisioning on a Range of PVCs

To enable ATM PVC autoprovisioning on a range of PVCs, enter the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.
Step 2	`Router(config-if)# range [range-name] pvc start-vpi/start-vci end-vpi/end-vci`	Specifies the range of PVCs and enters atm-range configuration mode.

	Command	Purpose
Step 3	`Router(config-if-atm-range)# create on-demand`	Enables PVC autoprovisioning on the range of PVCs.
Step 4	`Router(config-if-atm-range)# idle-timeout [time-out-in-seconds] [minimum-traffic-in-kbps]`	(Optional) Enables the idle-timeout timer on the on-demand PVC range. The default *time-out-in-seconds* is 0 (no idle-timeout). The Cisco 10000 series router waits until the traffic on a particular VC is processed before tearing down the VC, even if you specify the *minimum-traffic-in-kbps* option or if the VC is idle during the idle-timeout period.

Example 8-6 enables autoprovisioning on PVC range 100/100 to 100/200 and sets the idle-timeout timer for 300 seconds.

Example 8-6 Enabling ATM PVC Autoprovisioning on a PVC Range

```
Router(config)# int atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# range pvc 100/100 100/200
Router(config-subif-atm-range)# create on-demand
Router(config-subif-atm-range)# idle-timeout 300
```

Enabling ATM PVC Autoprovisioning on a Specific PVC Within a PVC Range

To enable ATM PVC autoprovisioning on a specific PVC within a PVC range, enter the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# interface atm slot/0 [.subinterface-number {multipoint | point-to-point}]`	Specifies the ATM interface and enters interface or subinterface configuration mode.
Step 2	`Router(config-if)# range [range-name] pvc start-vpi/start-vci end-vpi/end-vci`	Specifies the range of PVCs and enters atm-range configuration mode.
Step 3	`Router(config-if-atm-range)# pvc-in-range [pvc-name] [vpi/vci]`	Specifies an individual PVC within the PVC range.
Step 4	`Router(config-if-atm-range-pvc)# create on-demand`	Enables PVC autoprovisioning on the individual PVC within the PVC range.
Step 5	`Router(config-if-atm-range-pvc)# idle-timeout [time-out-in-seconds] [minimum-traffic-in-kbps]`	(Optional) Enables the idle-timeout timer on the on-demand PVC within the PVC range. The default *time-out-in-seconds* is 0 (no idle-timeout). The Cisco 10000 series router waits until the traffic on a particular VC is processed before tearing down the VC, even if you specify the *minimum-traffic-in-kbps* option or if the VC is idle during the idle-timeout period.

Example 8-7 enables autoprovisioning on PVC 100/100 in PVC range 100/100 to 100/200.

Example 8-7 Enabling ATM PVC Autoprovisioning on a PVC Within a PVC Range

```
Router(config)# int atm 3/0/0.1 multipoint
Router(config-subif)# atm pppatm passive
Router(config-subif)# range pvc 100/100 100/200
Router(config-subif-atm-range)# pvc-in-range 100/100
Router(config-subif-atm-range-pvc)# create on-demand
Router(config-subif-atm-range-pvc)# idle-timeout 300
```

Creating an On-Demand PVC With Infinite Range

To create an on-demand PVC with infinite range, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vc-class atm name`	Creates a VC class and enters vc-class configuration mode.
Step 2	`Router(config-vc-class)# create on-demand`	Enables PVC autoprovisioning. **Note** Configure additional VC parameters as appropriate. For more information, see the "VC Class" section on page 8-3 or the "Creating a VC Class Examples" section in the *Cisco IOS Wide-Area Networking Configuration Guide*.
Step 3	`Router(config-vc-class)# idle-timeout [time-out-in-seconds] [minimum-traffic-in-kbps]`	(Optional) Enables the idle-timeout timer on the on-demand PVC. The default *time-out-in-seconds* is 0 (no idle-timeout). The Cisco 10000 series router waits until the traffic on a particular VC is processed before tearing down the VC, even if you specify the *minimum-traffic-in-kbps* option or if the VC is idle during the idle-timeout period.
Step 4	`Router(config)# interface atm slot/0`	Specifies the ATM interface. You can configure infinite range only on a main ATM interface. When you use the **class-int** command to attach an ATM VC class to a subinterface, the **create on-demand** command is ignored.
Step 5	`Router(config-if)# class-int name`	Attaches a VC class to an ATM interface.

Example 8-8 creates a VC class named *myclass*, enables PVC autoprovisioning on the class, and attaches the class to the main ATM interface. All the PVCs on the main interface will be autoprovisioned; this configuration is also known as infinite range.

Example 8-8 Configuring an On-Demand PVC With Infinite Range

```
Router(config)# vc-class atm myclass
Router(config-vc-class)# create on-demand
Router(config-vc-class)# idle-timeout 300
Router(config)# int atm 3/0/0
Router(config-if)# class-int myclass
```

Monitoring and Maintaining ATM PVC Autoprovisioning

To monitor and maintain the ATM PVC autoprovisioning feature, enter any of the following commands in privileged EXEC mode.

Command	Purpose
`Router# show atm pvc`	Displays information about ATM PVCs, such as the interface, VPI/VCI, type, and encapsulation. PVC-A (PVC-Automatic) listed in the Type field indicates that the PVC is an on-demand PVC.
`Router# show atm vc`	Displays information about ATM VCs, including if the VC is an on-demand VC as indicated by VC-A (VC-Automatic) in the Type field.
`Router# show atm pvc` *VPI*`/`*VCI*	Displays information about a specific PVC, including if VC autoprovisioning is enabled.
`Router# debug atm autovc {event | error | all]`	Displays on-demand VC events and errors. Use the **event** option to display all on-demand VC events. Use the **error** option to display all on-demand VC errors. Use the **all** option to display both on-demand VC events and errors. **Note** Using the **debug atm autovc** command for a large range of PVCs can result in a large display of messages to the console window.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Example 8-9 indicates that autoprovisioning is enabled on PVCs 0/50, 0/51, and 0/52.

Example 8-9 show atm pvc Command

```
Router# show atm pvc

VCD /                                          Peak Avg/Min  Burst
Interface Name  VPI  VCI  Type  Encaps  SC  Kbps  Kbps    Cells  Sts
5/0.111    7     0    50   PVC-A  SNAP   UBR   149760             UP
5/0.111    8     0    51   PVC-A  SNAP   UBR   149760             UP
5/0.111    9     0    52   PVC-A  SNAP   UBR   149760             UP
```

Example 8-10 displays information about PVC 0/51 and indicates that autoprovisioning is enabled on the PVC.

Example 8-10 show atm pvc Command for a Specific PVC

```
Router# show atm pvc 0/51

ATM5/0.1: VCD: 118, VPI: 0, VCI: 51
UBR, PeakRate: 149760
AAL5-LLC/SNAP, etype:0x0, Flags: 0x20000C20, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s), OAM retry frequency: 1
second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP frequency: 15 minutes
Transmit priority 4
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InPRoc: 0, OutPRoc: 0, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0, LengthViolation: 0, CPIErrors: 0
Out CLP=1 Pkts: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OALM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Status: UP
PPP: Virtual-Access3 from Virtual-Template1
VC Auto Creation Enabled
```

Configuration Example for ATM PVC Autoprovisioning

The following example enables autoprovisioning on PVC range 100/100 to 100/3000 and applies the virtual template interface named *Virtual-Template1* to the PVC range.

```
ip local pool pool-1 10.1.1.2 10.1.12.255
!
interface Virtual-Template1
 ip address 10.1.1.1 255.255.0.0
 peer default ip address pool pool-1
 ip mtu 1492
 keepalive 60
 ppp timeout idle 65
 ppp direction callin
!
interface ATM7/0/0.1 point-to-multipoint
 atm pppatm passive
 range pvc 100/100 100/3000
  create on-demand
  idle-timeout 70
  encapsulation aal5mux ppp Virtual-Template1
```

Variable Bit Rate Non-Real Time Oversubscription

The Variable Bit Rate Non-Real Time (VBR-nrt) Oversubscription feature enables service providers to improve network utilization of otherwise underutilized shared networks by leveraging statistical multiplexing on ATM networks. Instead of supporting only unconditional reservation of network bandwidth to VCs, the router offers VC oversubscription to statistically guarantee bandwidth to VCs.

The VBR-nrt Oversubscription feature assumes that congestion at the physical port never or rarely occurs. For example, assume 10 VCs are configured to use 25 percent of the physical network bandwidth. The full capacity of the network is reached if only four VCs attempt to transmit traffic. The VBR-nrt Oversubscription feature is intended only for networks with low utilization in which congestion is unlikely to exist.

In releases earlier than Cisco IOS Release Cisco IOS Release 12.3(7)XI1, a call admission check (CAC) prevented you from assigning more bandwidth to virtual circuits (VCs) than a port's total bandwidth. The Cisco 10000 series router supported unconditional reservation of network bandwidth to VCs. When the sum of the transmission capacities of VCs falls within the bandwidth of the physical network, the network does not congest. Each VC receives its bandwidth reservation regardless of the traffic pattern of any other VC on that network. However, VCs receive this unconditional service at the expense of underutilization of the physical capacity of the network. Because each VC uses a fraction of the physical capacity, unless a large number of VCs remain busy, the overall network utilization remains low.

In Cisco IOS Release Cisco IOS Release 12.3(7)XI1 or later, the VBR-nrt Oversubscription feature enables you to specify the amount of oversubscription (oversubscription factor) you want to allow. The CAC check is based on the oversubscription factor you specify and evaluated separately for both VCs and VP tunnels into the port, and VCs into VP tunnels.

The oversubscription factor is also used to evaluate the amount of bandwidth allocated for unspecified bit rate (UBR) VCs. Prior to Cisco IOS Release Cisco IOS Release 12.3(7)XI1, UBR VCs received the bandwidth remaining after other VCs had been allocated bandwidth. The CAC check now adjusts the bandwidth for UBR VCs based on the oversubscription factor. For example:

port speed- sum of the VBR VCs = aggregate UBR bandwidth

Note You can apply a nested policy map to the main ATM interface to override this default equation and set a specific bandwidth for the aggregate UBR queues.

Whenever the oversubscription factor is reduced, less bandwidth is available for VC creation. As a result, a warning message appears indicating that some VCs might not be created. The VCs are not explicitly removed from the configuration and remain up and functional until you reboot the router or reset the slot. At this point, the VCs remain in the configuration but they are not up.

For optimal performance, configure the oversubscription factor as closely as possible to the sum of all VCs. The system allows VCs to be added provided the total subscribed rate is less than or equal to the port speed * over-subscription-factor.

CAC checks are disabled when the **no atm pxf queuing** command is configured on an interface.

For QoS-related information, see Chapter 13, "Oversubscribing Physical and Virtual Links" in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

The VBR-nrt Oversubscription feature is described in the following topics:

Feature History for VBR-nrt Oversubscription

Cisco IOS Release	Description	Required PRE
12.2(16)BX3	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for VBR-nrt Oversubscription

The VBR-nrt Oversubscription feature has the following restrictions:

Congestion

- Due to congestion on the physical interface, the accuracy of priority queuing (PQ) and class-based weighted fair queuing (CBWFQ) on individual VCs degrades. For example, if you configure each of three queues at a distribution of 50, 30, and 20 percent, the actual distribution might be 45, 40, and 15 percent.
- The distribution of bandwidth for each VC might be less than expected based on the speed of the VC. Typically, low speed VCs are allocated the expected bandwidth while high speed VCs share the remaining bandwidth equally.
- The amount of bandwidth allocated for the PQ or latency might be less than expected.

Oversubscription Feature

- Oversubscription of the ATM interfaces is off by default. Oversubscription of the tunnels (the number and bandwidth of VCs that can be in a tunnel) is on by default and is not subject to any oversubscription factor. Oversubscription of the tunnels cannot be adjusted or turned off.
- Use the **atm over-subscription-factor** command to enable the oversubscription feature for a particular interface or tunnel.

Note Do *not* use the **atm oversubscribe** command to enable oversubscription, as this can cause undesirable results.

The following configuration enables the oversubscription feature and configures the interface with an over-subscription-factor of 50.

```
Router(config)# interface atm 4/0/0
Router(config-if)# atm over-subscription-factor 50
Router(config-if)# exit
```

- To prevent oversubscription of the interface, use the **no atm oversubscribe** command. For example, the following configuration disables oversubscription of the ATM 4/0/0 interface. The previously configured factor 50 is configured on the interface, but the router does not allow the oversubscription.

```
Router(config)# interface atm 4/0/0
Router(config-if)# no atm oversubscribe
Router(config-if)# exit
```

- It is recommended that the **atm over-subscription-factor** command be applied to all ports of an ATM line card. This command controls the allocation of resources that are managed on a line card. Enabling oversubscription on one port alone could result in other ports taking up more resources than they were supposed to use. This could result in starving other ports for resources, which could cause VC creation to fail.
- In **atm pxf queuing** mode, the number of active VCs the ATM line cards support for Cisco IOS Release 12.3(7)XI2 or later releases is shown in Table 8-2.

Table 8-2 Active VCs on ATM Line Cards

Line Card	Max. VCs per Port	Maximum VCs per Module	No. VBR, CBR, Shaped UBR VCs
E3/DS3	4,096	32,768[1]	28,672[2]
OC-3	8,191	32,764[3]	28,672[4]
OC-12	16,384 (previously 14,436)	16,384	16,384

1. For 32,768 VCs per module, 4096 of them must be unshaped UBR VCs.
2. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.
3. For 32,764 VCs per module, 4096 of them must be unshaped UBR VCs.
4. For 28,672 VBR, CBR, and shaped UBR VCs, no VCs can be in shaped VP tunnels. If VCs are in shaped VPs, the number of VBR, CBR, and shaped UBR VCs is 22,204.

You can configure the maximum number of VCs across the ports in any fashion, provided that you do not exceed the per-port maximum.

Although the maximum number of VBR, CBR, and shaped UBR VCs per E3/DS3 and OC-3 ATM line card is 28,672 VCs, the router supports a maximum of 22,204 VBR, CBR, and shaped UBR VCs per line card that you can place within virtual path (VP) tunnels. If you attempt to bring up more than 22,204 VCs in a configuration that includes VP tunnels and VCs (hierarchical traffic shaping configuration), the VCs might not assign traffic correctly or the VCs might not come up at all. Be sure to limit the number of configured VBR, CBR, and shaped UBR VCs on an ATM card to less than 22,204 VCs if you place the VCs in VP tunnels.

For the OC-12 ATM line card, the router supports 16,384 VCs in VP tunnels.

Configuration Tasks for VBR-nrt Oversubscription

To configure the VBR-nrt Oversubscription feature, perform the following configuration tasks:

- Configuring VBR-nrt Oversubscription, page 8-17
- Verifying ATM PVC Oversubscription, page 8-17

Configuring VBR-nrt Oversubscription

To enable oversubscription of ATM VCs, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# atm over-subscription-factor {1-500}`	Oversubscribes an ATM VC. *1-500* specifies the amount of oversubscription. The default value is 1. **Note** Use this command for each ATM interface that you want to oversubscribe.

Note You do not need to use the **service-policy** command to specify the ATM VC oversubscription because a variable bit rate (VBR) ATM VC uses sustained cell rate (SCR) to define the VC average transmission rate.

Example 8-11 oversubscribes an ATM interface by five times the physical transmission capacity.

Example 8-11 Oversubscribing an ATM VC

```
Router(config)# interface atm 4/0/0
Router(config-if)# atm over-subscription-factor 5
```

Verifying ATM PVC Oversubscription

To verify the configuration of ATM PVC oversubscription, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show controllers interface`	Displays the total subscribed rate of all VCs on the port. The system allows VCs to be added provided the total subscribed rate is less than or equal to: port speed * over-subscription-factor.
`Router# show running-config`	Displays the contents of the currently running configuration file. Indicates that oversubscription is on.

Configuration Example for ATM PVC Oversubscription

The following example oversubscribes an ATM interface by 10 times the physical transmission capacity:

```
interface atm 4/0/0
 atm over-subscription-factor 10
```

CHAPTER 9

Configuring Multihop

In a Virtual Private Dialup Network (VPDN) environment, sessions generated from a remote host are routed over an existing tunnel or a tunnel built to route a specific domain. Typically, sessions cannot traverse more than one L2TP tunnel before reaching the ISP or corporate network. However, by using the Multihop feature, you can configure the Cisco 10000 series router to terminate sessions arriving in L2TP tunnels from a LAC and then route the remote traffic through new L2TP tunnels to an LNS device in the ISP or corporate network.

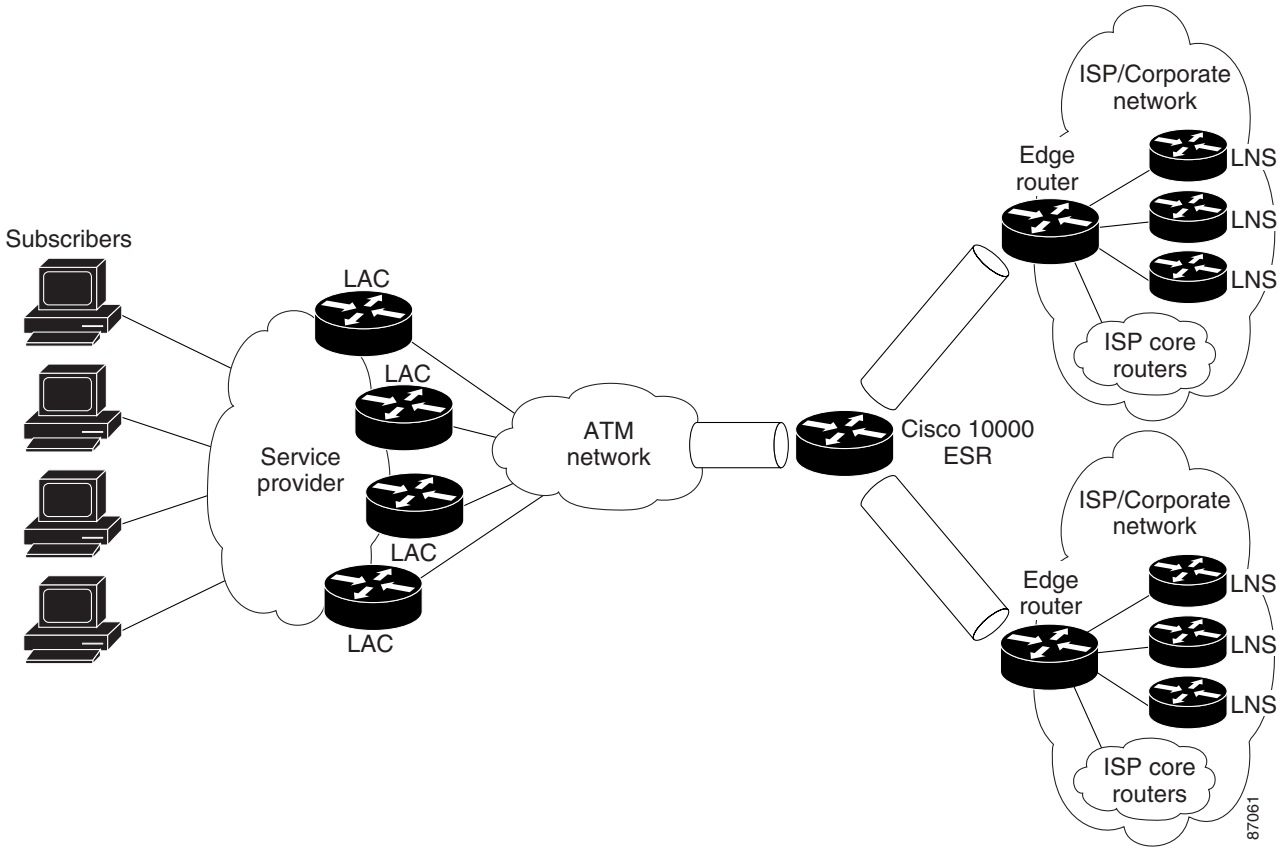
The Multihop feature enables the Cisco 10000 series router to terminate sessions arriving in L2TP tunnels from a LAC and to forward the sessions through new L2TP tunnels to the router's peer L2TP Network Server (LNS). The packets arrive at the router with L2TP encapsulation and the router forwards the packets with a different L2TP encapsulation. The Cisco 10000 router maps the sessions to the new tunnels based on the session's domain or the tunnel in which the session arrived.

The Cisco 10000 router also supports the preservation of the IP type of service (TOS) field for tunneled IP packets. Each L2TP data packet and IP packet has a TOS field. When the router creates an L2TP data packet, the TOS field sets to zero (normal service), ignoring the TOS field of the encapsulated IP packet being tunneled. To preserve quality of service for tunneled packets, the Cisco 10000 router supports the configuration of accept-dialin and request-dialout VPDN groups using the **l2tp ip tos reflect** command. When the router creates an L2TP data packet at a virtual-access interface (VAI), instead of ignoring the IP packet TOS field, the router copies the field onto the L2TP data packet.

Note Typically, the Cisco IOS software reflects the TOS field from the inner packet header to the outer packet header. However, the Cisco 10000 router propagates the TOS field from the ingress header to the egress header.

Figure 9-1 shows an example of a multihop topology. On the access network side, the Cisco 10000 router connects to access provider LACs. On the provider network side, the router connects to LNS devices in other ISP or corporate provider networks. Multiple L2TP tunnels are carried over either multiple interfaces or a single interface. Typically, the connection between the router and the LAC or the router and the LNS is an ATM connection. However, this is not a requirement. You can use any interface that can carry L2TP tunneled traffic.

Figure 9-1 Multihop Topology Example



This chapter describes the Multihop feature in the following topics:

Feature History for Multihop

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for Multihop

The Multihop feature has the following restrictions:

- The performance routing engine, part number ESR-PRE1 does not support the Multihop feature.
- Tunnel switching is based on a session's domain or tunnel in which the session arrived. The Cisco 10000 router does not support switching of individual sessions by using the CLI.
- The Cisco 10000 router does not support multichassis Multilink PPP (MLPPP).
- The Cisco 10000 router supports the Multihop feature for L2TP, but does not support the L2F protocol.
- You cannot apply per session features to switched sessions. For example, you cannot apply an ACL or a service policy to the sessions.

To preserve the IP TOS field of tunneled IP packets, the following restrictions apply:

- The Cisco 10000 router supports only the L2TP tunneling protocol.
- The tunneled link must carry IP to preserve the TOS field.
- The Cisco 10000 router does not support proxy PPP dialin.

Required Configuration Tasks for Multihop

To configure the Multihop feature on the Cisco 10000 router, perform the following configuration tasks:

- Enabling VPDN and Multihop Functionality, page 9-3
- Terminating the Tunnel from the LAC, page 9-4
- Mapping the Ingress Tunnel Name to an LNS, page 9-4

Enabling VPDN and Multihop Functionality

To enable VPDN and multihop functionality, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN functionality.
Step 2	`Router(config)# vpdn multihop`	Enables VPDN multihop functionality.

Terminating the Tunnel from the LAC

To terminate the tunnel from the LAC, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username remote-hostname password secret`	Configures the secret (password) for the remote LAC. The *secret* must match the *secret* configured on the LAC and can consist of any string of up to 11 ASCII characters.
Step 2	`Router(config)# username local-name password secret`	Configures the secret (password) for the local device. The *secret* must match the *secret* configured in step 1.
Step 3	`Router(config)# vpdn-group number`	Selects the VPDN group.
Step 4	`Router(config-vpdn)# accept-dialin`	Accepts tunneled PPP connections from the LAC and creates an accept-dialin virtual private dialup network (VPDN) subgroup.
Step 5	`Router(config-vpdn-acc-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol (L2TP) that the VPDN subgroup will use.
Step 6	`Router(config-vpdn-acc-in)# virtual-template number`	Specifies the virtual template interface to use to clone the new virtual access interface.
Step 7	`Router(config-vpdn-acc-in)# exit`	Returns to VPDN group mode.
Step 8	`Router(config-vpdn)# terminate-from hostname remote-hostname`	Specifies the host name of the remote LAC that is required when accepting a VPDN tunnel. The *remote-hostname* must match the *remote-hostname* configured in Step 1.
Step 9	`Router(config-vpdn)# local name local-name`	Specifies the local host name that the tunnel will use to identify itself. The *local-name* must match the *local-name* configured in Step 2.

Mapping the Ingress Tunnel Name to an LNS

To map the ingress tunnel name to an LNS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username username password secret`	Configures the secret (password) for the LNS. The *username* must match the LNS hostname or tunnel ID. The *secret* must match the *secret* configured on the LNS.
Step 2	`Router(config)# username egress-tunnel-name password secret`	Configures the secret (password) for the tunnel. The *egress-tunnel-name* specifies the remote (LNS) host name of the tunnel. The *secret* must match the *secret* configured in Step 1.
Step 3	`Router(config)# vpdn-group number`	Selects the VPDN group and enters VPDN configuration mode.
Step 4	`Router(config-vpdn)# request-dialin`	Enables the Cisco 10000 router to request L2TP tunnels to the LNS and enters VPDN request-dialin subgroup mode.
Step 5	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol (L2TP) that the VPDN subgroup will use.

	Command	Purpose
Step 6	`Router(config-vpdn-req-in)# multihop hostname ingress-tunnel-name`	Initiates a tunnel based on the LAC's hostname or ingress tunnel ID.
Step 7	`Router(config-vpdn-req-in)# exit`	Returns to VPDN group mode.
Step 8	`Router(config-vpdn)# initiate-to ip ip-address [limit limit-number] [priority priority-number]`	Specifies the IP address of the LNS that will be tunneled to. Optionally, you can configure the maximum number of connections that can be made to the IP address and the priority for the IP address (1 is the highest).
Step 9	`Router(config-vpdn)# local name egress-tunnel-name`	Specifies the local host name that the tunnel uses to identify itself. The *egress-tunnel-name* must match the *egress-tunnel-name* configured in Step 2.

Optional Configuration Tasks for Multihop

To configure the Multihop feature on the Cisco 10000 router, perform any of the following optional tasks:

- Specifying VPDN Tunnel Authorization Searches by Ingress Tunnel Name, page 9-5
- Preserving the Type of Service Field of Encapsulated IP Packets, page 9-5

Specifying VPDN Tunnel Authorization Searches by Ingress Tunnel Name

To specify that the provider's network access server is to perform VPDN tunnel authorization searches by using the ingress tunnel name, enter the following command in global configuration mode:

Command	Purpose
`Router (config)# vpdn search-order multihop-hostname [domain]`	Specifies a search by the configured ingress tunnel name. Optionally, you can specify to search by domain name only.

Preserving the Type of Service Field of Encapsulated IP Packets

To preserve the type of service (TOS) field of encapsulated IP packets, perform the following configuration tasks:

- Configuring an Accept-Dialin VPDN Group to Preserve IP TOS, page 9-6
- Configuring a Request-Dialout VPDN Group to Preserve IP TOS, page 9-7

Configuring an Accept-Dialin VPDN Group to Preserve IP TOS

To configure an accept-dialin VPDN group to preserve IP TOS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group number`	Selects the VPDN group and enters VPDN configuration mode.
Step 2	`Router(config-vpdn)# accept-dialin`	Accepts tunneled PPP connections from the LAC and creates an accept-dialin virtual private dialup network (VPDN) subgroup.
Step 3	`Router(config-acc-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol (L2TP) that the VPDN subgroup will use. **Note** L2TP is the only protocol that supports dialout and IP TOS preservation.
Step 4	`Router(config-vpdn-acc-in)# virtual-template number`	Specifies the virtual template interface to use to clone the new virtual access interface.
Step 5	`Router(config-vpdn-acc-in)# exit`	Returns to VPDN group mode.
Step 6	`Router(config-vpdn)# terminate-from hostname remote-hostname`	Specifies the host name of the remote LAC that will be required when accepting a VPDN tunnel.
Step 7	`Router(config-vpdn)# local name local-name`	Specifies the local host name that the tunnel will use to identify itself.
Step 8	`Router(config-vpdn)# ip tos reflect`	Configures the VPDN group to preserve the TOS field of L2TP tunneled IP packets.

Example 9-1 configures *vpdn-group 1* to accept tunneled PPP connections from the remote LAC named *myhost* and to preserve the TOS field of L2TP tunneled IP packets.

Example 9-1 Configuring an Accept-Dialin VPDN Group for IP TOS Preservation

```
vpdn-group 1
    accept-dialin
        protocol l2tp
        virtual-template 1
    terminate-from hostname myhost
    local name local-host1
    ip tos reflect
```

Configuring a Request-Dialout VPDN Group to Preserve IP TOS

To configure a request-dialout VPDN group to preserve IP TOS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group number`	Selects the VPDN group and enters VPDN configuration mode.
Step 2	`Router(config-vpdn)# request-dialout`	Enables the LNS to request L2TP tunnels for dialout calls.
Step 3	`Router(config-vpdn-req-out)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol (L2TP) that the VPDN subgroup will use. **Note** L2TP is the only protocol that supports dialout and IP TOS preservation.

Step 4	`Router(config-vpdn-req-out)# pool-member pool-number`
	OR
	`Router(config-vpdn-req-out)# rotary-group group-number`
		Specifies the dialer profile pool or dialer rotary group to use to dial out.
		Note You can only configure one dialer profile pool or one dialer rotary group. Attempting to configure a second dialer resource removes the first resource from the configuration.

Step 5	`Router(config-vpdn-req-out)# exit`	Returns to VPDN group mode.
Step 6	`Router(config-vpdn)# initiate-to ip ip-address [limit limit-number] [priority priority-number]`	Specifies the IP address of the LNS that is dialed out. Optionally, you can configure the maximum number of connections that can be made to the IP address and the priority for the IP address (1 is the highest).
Step 7	`Router(config-vpdn)# local name local-name`	Specifies the local host name that the tunnel uses to identify itself.
Step 8	`Router(config-vpdn)# ip tos reflect`	Configures the VPDN group to preserve the TOS field of L2TP tunneled IP packets.

Example 9-2 configures *vpdn-group 1* for L2TP dialout tunnel preservation of the IP TOS.

Example 9-2 Configuring a Request-Dialout VPDN Group for IP TOS Preservation

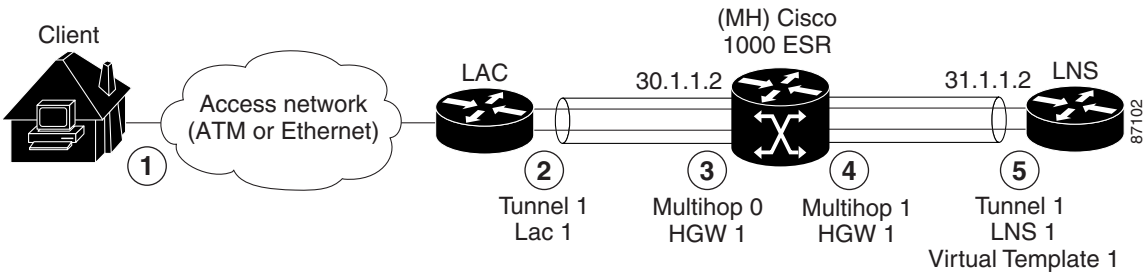
```
vpdn-group 1
    request-dialout
        protocol l2tp
        pool-member 1
    initiate-to ip 10.16.49.94
    ip tos reflect
```

Configuration Examples for Multihop

The example in this section is a multihop configuration in which the Cisco 10000 router is configured as the multihop system (MH). The example includes LAC and LNS configurations to complete the configuration. This configuration scenario supports a maximum of two hops between the LAC device and the destination LNS device.

Figure 9-2 shows the example multihop configuration, described in more detail in the list that follows.

Figure 9-2 Multihop Configuration Example



The remote client dials in to the LAC. The LAC negotiates link control protocol (LCP) and preauthenticates the user.

5. The LAC configuration sets up a vpdn-group named *tunnel1*. This vpdn-group initiates a tunnel to IP address 30.1.1.2 to request dialin connection for any packets associated with the *cisco.com* domain. The local name of tunnel1 is *LAC1*. This is the name by which tunnel1 identifies itself to the receiving end of the L2TP tunnel.
6. The Cisco 10000 router acts as the multihop system (MH). On the LAC side, the MH configuration requires users to log in to the system. The MH configuration creates a vpdn-group named *multihop0*, which identifies the L2TP tunnel terminating from the LAC. The multihop0 tunnel only accepts dialin connections from the LAC and identifies itself by using the local name *Home Gateway 1* (*HGW1*).
7. On the LNS side, the MH configuration creates a vpdn-group named *multihop1*, which initiates an L2TP tunnel to the LNS at IP address 31.1.1.2. The multihop1 vpdn-group requests dialin connections to the LNS based on the LAC's hostname. Using the **multihop hostname LAC1** command creates the association between the LAC and the LNS devices. Like multihop0, multihop1 shares the same HGW1 local name.
8. The LNS configuration sets up a vpdn-group named *tunnel1*, which accepts dialin connections from the MH system. The tunnel1 vpdn-group terminates the L2TP tunnel from the MH system (identified by the HGW1 local name) and uses the local name *LNS1* to identify itself. The LNS configuration creates a virtual template interface named Virtual-Template1, which it associates with tunnel1. Virtual-Template1 uses PAP authentication and assigns the IP address by using the local IP address pool named *pool-1*.

LAC Configuration

```
!
vpdn enable
!
vpdn-group tunnel1
 request-dialin
  protocol l2tp
  domain cisco.com
 initiate-to ip 30.1.1.2  priority 1
```

```
 local name LAC1
 l2tp tunnel password 7 060A0E23
 l2tp tunnel receive-window 100
 l2tp tunnel retransmit timeout min 2
!
```

Multihop Configuration

```
username user@cisco.com password 0 lab
!
vpdn enable
vpdn multihop
vpdn search-order multihop-hostname domain dnis
!
vpdn-group multihop0
 accept-dialin
  protocol l2tp
 terminate-from hostname LAC1
 local name HGW1
 l2tp tunnel password 7 09404F0B
!
vpdn-group multihop1
 request-dialin
  protocol l2tp
  multihop hostname LAC1
 initiate-to ip 31.1.1.2  priority 1
 local name HGW1
 l2tp tunnel password 7 0507070D
!
```

LNS Configuration

```
vpdn enable
!
vpdn-group tunnel1
 accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname HGW1
 local name LNS1
 l2tp tunnel password 7 04570A04
 l2tp tunnel receive-window 100
 l2tp tunnel retransmit timeout min 2
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet2/0/0
 no keepalive
 peer default ip address pool pool-1
 ppp mtu adaptive
 ppp authentication pap callin
!
ip local pool pool-1 4.2.0.0 4.2.255.255
```

Monitoring and Maintaining Multihop Configurations

To monitor and maintain multihop configurations and VPDN groups, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the current router configuration. Use the output of this command to ensure that the configuration: • Enables VPDN and multihop functionality • Terminates tunnels from the LAC • Maps the ingress tunnel name to the LNS • Performs VPDN tunnel authorization searches by ingress tunnel name • (Optional) Configures an accept-dialin and request-dialout VPDN group to preserve the TOS field of L2TP tunneled IP packets
`Router# show vpdn`	Displays information about active L2TP tunnels and sessions.
`Router# show vpdn session [all [interface | tunnel | username] | packets | sequence | state | timers | window]`	Displays VPDN session information including interface, tunnel, username, packets, status, and window statistics.
`Router# show vpdn tunnel [all [id | local-name | remote-name] | packets | state | summary | transport]`	Displays VPDN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.
`Router# show interface virtual-access number`	Displays information about the virtual access interface, LCP, protocol states, and interface statistics. The following information indicates a normal working status for the virtual access interface (# indicates the number of the VAI): `Virtual-Access# is up, line protocol is up`
`Router# clear vpdn tunnel [l2tp [remote-name | local-name]]`	Shuts down a specific tunnel and all the sessions within the tunnel.
`Router# debug vpdn event [protocol | flow-control]`	Displays VPDN errors and basic events within the L2TP protocol. Also displays errors associated with flow control. **Note** Flow control is only possible if you use L2TP and you configure the remote peer receive window with a value greater than zero.
`Router# debug vpdn error`	Displays errors that prevent a tunnel from being established or errors that cause an established tunnel to be closed.
`Router# debug vpdn packet [control | data] [detail]`	Displays protocol-specific packet header information, such as sequence numbers, flags, and length.
`Router# debug vpdn 12x-events`	Displays L2TP events that are part of tunnel establishment or shutdown.
`Router# debug vpdn 12x-errors`	Displays L2TP protocol errors that prevent tunnel establishment or normal operation.
`Router# debug vpdn 12x-packets`	Displays the dialog between the LAC and LNS for tunnel or session creation.
`Router# debug vpdn 12x-data`	Checks L2TP data transfer.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Example 9-3 shows the information that displays when you use the **show vpdn** command. All tunnel and session information displays for all active sessions and tunnels when you use the **show vpdn** command without any keywords or arguments.

Example 9-3 show vpdn Command

```
Router# show vpdn
L2TP Tunnel and Session Information Total tunnels 2 sessions 22
LocID RemID Remote Name   State  Remote Address  Port  Sessions VPDN Group
1206019602tunnel5est45.1.5.5170111tunnel5

LocIDRemIDTunIDIntfUsernameStateLast Chg
3  312060SSSCircuitu@n5est2d19h
2  212060SSSCircuitu@n5est2d19h
4  412060SSSCircuitu@n5est2d19h
5  512060SSSCircuitu@n5est2d19h
6  612060SSSCircuitu@n5est2d19h
7  712060SSSCircuitu@n5est2d19h
8  812060SSSCircuitu@n5est2d19h
9  912060SSSCircuitu@n5est2d19h
10 1012060SSSCircuitu@n5est2d19h
11 1112060SSSCircuitu@n5est2d19h
12 1212060SSSCircuitu@n5est2d19h

LocID RemID Remote Name   State  Remote Address  Port  Sessions VPDN Group
103352883tunnel6est45.1.6.5170111tunnel6

LocIDRemIDTunIDIntfUsernameStateLast Chg
14 1410335SSSCircuitu@n6est2d19h
15 1510335SSSCircuitu@n6est2d19h
16 1610335SSSCircuitu@n6est2d19h
17 1710335SSSCircuitu@n6est2d19h
18 1810335SSSCircuitu@n6est2d19h
19 1910335SSSCircuitu@n6est2d19h
20 2010335SSSCircuitu@n6est2d19h
21 2110335SSSCircuitu@n6est2d19h
22 2210335SSSCircuitu@n6est2d19h
23 2310335SSSCircuitu@n6est2d19h
13 1310335SSSCircuitu@n6est2d19h

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels
```

Example 9-4 uses the **show interface virtual-access** command to display information about virtual access interface 3. In this example, the following information indicates a normal working status:

```
Virtual-Access3 is up, line protocol is up
```

Example 9-4 show interface virtual access Command

```
Router# show interface virtual-access 3
Virtual-Access3 is up, line protocol is up
    Hardware is Virtual Access interface
    MTU 1500 bytes, BW 128 Kbit, DLY 100000 usec, rely 255/255, load 1/255
    Encapsulation PPP, loopback not set, keepalive set (10 sec)
    DTR is pulsed for 5 seconds on reset
    LCP Open, multilink Open
    Open: IPCP
    Last input 00:02:30, output never, output hang never
    Last clearing of "show interface" counters 1d19h
    Queueing strategy: fifo
    Output queue 0/40, 0 drops; input queue 21/75, 0 drops
    5 minute input rate 0 bits/sec, 0 packets/sec
    5 minute output rate 0 bits/sec, 0 packets/sec
        55930 packets input, 3347967 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        105261 packets output, 9607052 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 output buffer failures, 0 output buffers swapped out
        0 carrier transitions
```

CHAPTER 10

Configuring Address Pools

Service providers concerned with the efficient management of IP address space are challenged to implement an address assignment mechanism that efficiently assigns addresses to remote users from address pools and effectively manages those pools. Such deployment requires a strategy for dealing with poorly utilized address pools and pools that run out of addresses. Each remote user assigned an address must have a route to the remote user configured in the corresponding virtual routing and forwarding (VRF) instance. Configuration becomes further complicated by the fact that a single PE router can support hundreds of VRFs, and the provider's network can have hundreds or thousands of PE routers. The total number of routes in all VRFs and in the default routing table on a single PE router can grow enormously, highlighting the need for an address mechanism that provides for route summarization.

To enhance IP address space management, the Cisco 10000 series router supports the following address pool features:

- On-Demand Address Pool Manager, page 10-4—Provides an address assignment mechanism that dynamically resizes address pools and permits efficient route summarization.
- Overlapping IP Address Pools, page 10-16—Enables you to use multiple IP address spaces and reuse IP addresses among different VPNs supported on the Cisco 10000 series router.

This chapter describes the advantages and disadvantages of address assignment mechanisms currently deployed, the On-demand Address Pool Manager feature, and the Overlapping IP Address Pools feature:

Address Assignment Mechanisms

Typically, service providers deploy the following address assignment mechanisms:

The following sections describe the advantages and disadvantages of the address assignment mechanisms.

Local Address Pool

A local address pool is a pool of IP addresses statically configured on a PE router. The pool name identifies the address pool. When a PPP session requests an address from a specific pool, the pool manager assigns an unused address from the pool. When the PPP session returns the address, the pool manager puts the address back into the pool from which it was taken.

A common group identifier identifies a group of pools. In an MPLS VPN network architecture, each pool group is used to assign addresses to remote users belonging to a particular VPN. Though not officially associated with a VRF, the address pool is unofficially tied to the VRF because each VPN associated with an address pool is also associated with a specific VRF.

The ability to assign overlapping addresses provides a significant benefit to VPN customers who use private addresses. Two address pools in different groups can have overlapping IP addresses, but two pools in the same group cannot contain overlapping addresses.

Benefits of a Local Address Pool

The main benefit of a local address pool is the ability to efficiently summarize routes:

- The total number of routes configured on a single PE router can grow enormously. Route summarization avoids lengthy VRF and default routing tables.
- Summarized routes correspond to all subnets present in the address pool.
- The summarized routes are configured in the VRF associated with the address pool.

Limitations of a Local Address Pool

A drawback to local address pools is that because they are statically configured, the pool might be poorly utilized or it might run out of addresses. The provider's ISP customers have a limited number of public addresses and are particularly affected by poorly managed pools. For example, for the same ISP it is possible that one PE router is underutilizing its local pool while another PE router has exhausted its local pool.

RADIUS-Based Address Assignment

RADIUS is a distributed client/server system that secures networks against unauthorized access. In addition to providing authentication, authorization, and accounting (AAA) services, RADIUS also provides IP address assignment by using user defined static routes and IP pool definitions on the RADIUS server.

In the Cisco 10000 series router implementation, a RADIUS client runs on the router and queries a central RADIUS server for a remote user's static route or an IP address from the RADIUS IP pool definitions. Typically, the RADIUS server assigns addresses from a separate pool of addresses for each VPN associated with a particular PE router. This allows the server to assign contiguous addresses to remote users who are in the same VPN and who connect to the same PE router. The RADIUS server uses the remote user's domain name to identify the VPN.

Benefits of RADIUS-Based Address Assignment

RADIUS is an effective mechanism for providing IP address assignment for remote users:

- One benefit of RADIUS-based address assignment is its ability to effectively manage the IP address pools configured on the server. RADIUS can dynamically resize pools as needed, removing addresses from poorly utilized pools and adding them to pools that run out of addresses.
- RADIUS supports route summarization and uses profiles configured on the server to provide efficient addressing and AAA services.
- RADIUS can also attach a fixed IP address to a remote user's login.

Limitations of RADIUS-Based Address Assignment

When deciding upon an addressing mechanism, you must weigh the limitations and benefits of RADIUS-based address assignment. The following are some of the limitations of RADIUS-based address assignment:

- Using RADIUS for address assignment can increase the load on the server and slow the server's performance.
- As remote users log on and off, route summarization can become less efficient because it becomes more difficult for the PE router to have a contiguous set of IP addresses that the PE can summarize to the RADIUS server.
- Each time a user logs on or off, the Border Gateway Protocol (BGP) sends update information to the PE routers to update the VRFs configured on each router.
- Remote users have limited connectivity during the time it takes for BGP to propagate a newly configured route to all PE routers.

DHCP-Based Address Assignment

Dynamic Host Configuration Protocol (DHCP) servers allocate IP addresses to remote users, eliminating the need to configure users individually. DHCP also provides all the parameters that user systems require to operate and exchange information on the Internet network to which they connect. DHCP is based on a client/server model. The client software runs on the user system and the server software runs on the DHCP server.

DHCP uses a lease mechanism that offers an automated, reliable, and safe method for distributing and reusing addresses in networks, with little need for administrative intervention. As a system administrator, you can tailor the lease policy to meet the specific needs of your network.

Leases are grouped together in an address pool called a *scope*. The scope defines the set of IP addresses available for requesting hosts. A lease can be reserved (the host always receives the same IP address) or dynamic (the host receives the next available, unassigned lease in the scope).

Benefits of DHCP-based Address Assignment

One of the most significant benefits of DHCP is that it can dynamically configure user systems with IP addresses and associate leases with the assigned addresses. DHCP also provides for multiple servers. You can configure redundant DHCP servers so that if one server cannot provide leases to requesting clients, the other one can take over. Existing DHCP clients can continue to keep and renew their leases without knowing which server is responding to their requests.

Limitations of DHCP-Based Address Assignment

DHCP-based address assignment has route summarization problems similar to the problems encountered with RADIUS-based address assignment. Route summarization becomes less efficient as remote users log on and off, and users have limited connectivity while BGP updates all of the PE routers with newly configured routes.

For more information, see the "RADIUS-Based Address Assignment" section on page 10-2.

On-Demand Address Pool Manager

The On-demand Address Pool Manager feature is a mechanism for assigning and managing IP addresses.

On-demand address pools (ODAPs) use a central server to manage a block of addresses for each customer. The central server can be a Dynamic Host Configuration Protocol (DHCP) server or a RADIUS server. After the ODAP is configured, the central server populates the ODAP with one or more subnets leased from the central server. The central server divides each address pool into subnets and assigns the subnets to PE routers upon request.

Note The Cisco Network Registrar (CNR) DHCP server and the Cisco Access Registrar (CAR) RADIUS server support ODAPs.

The customer site connects to a provider edge (PE) router in the provider network. When an ODAP is configured, the pool manager for the PE router initiates a request to the central server for an initial subnet for a specific ODAP. The pool manager then monitors the utilization of the ODAP.

If the utilization of the pool exceeds a high-utilization threshold (high-utilization mark), the pool manager requests an additional subnet from the central server and adds it to the ODAP. Similarly, if the utilization of the pool decreases below the low-utilization threshold (low-utilization mark), the pool manager returns one or more subnets to the central server from which it was originally leased. Each time subnets are added to or removed from the ODAP, the summarized routes for each leased subnet must be inserted or removed from the corresponding VRF.

The On-demand Address Pool Manager feature is described in the following topics:

Feature History for On-Demand Address Pool Manager

Cisco IOS Release	Description	Required PRE
12.2(15)BX	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Address Allocation for PPP Sessions

For individual address allocation for PPP sessions, the pool manager searches for a free address beginning with the very first leased subnet. If a free address is not available in the first subnet, the pool manager searches the second leased subnet, and so on until a free address is found. This method of address allocation allows for efficient subnet release and route summarization. However, it differs from the normal DHCP address selection policy in which the IP address of the receiving interface is taken into account. The on-demand address pool manager feature provides an IP address pooling mechanism for PPP that allows the DHCP server to distinguish between a normal DHCP address request and an address request for a PPP client.

Subnet Releasing

The pool manager releases subnets beginning with the last leased subnet. The pool manager searches for a releasable subnet—a subnet with no addresses currently being leased. If it finds a releasable subnet, it releases the subnet and removes the summarized route for that subnet. If more than one releasable subnet exists, the pool manager releases the most recently allocated subnet. The pool manager takes no action if it does not find a releasable subnet. If the high utilization mark is reached by releasing the subnet, the pool manager does not release the subnet. Regardless of the instantaneous utilization level, the pool manager never releases the first leased subnet until it disables the ODAP.

On-Demand Address Pools for MPLS VPNs

The on-demand address pool manager feature provides support for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) environments. This feature automates the resizing of address pools, reducing network loading and manual configuration.

Each ODAP is configured and associated with a specific MPLS VPN. Each VPN is associated with one or more VRFs. The VRF maintains a routing table and other information associated with a specific customer VPN site. The utilization of the on-demand address pool occurs as described in the "On-Demand Address Pool Manager" section on page 10-4, except that address allocation occurs within the VRF associated with a particular VPN.

Only the ODAP associated with a specific VPN can allocate addresses to PPP sessions belonging to that VPN. The PE router on which the ODAP is configured terminates the PPP sessions and maps the remote user to the corresponding MPLS VPNs.

Note For more information about ODAPs, see the "On-Demand Address Pool Manager" section on page 10-4. For information about configuring MPLS VPNs, see the Remote Access to MPLS VPN chapter or see the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.

The On-demand Address Pools for MPLS VPNs feature is described in the following topics:

Benefits On-Demand Address Pool Manager

The on-demand address pool manager feature provides:

- Dynamic resizing of IP address pools, increasing or reducing the size of the pool as needed
- Automated control of address assignment
- Easy monitoring capabilities, enabling the pool manager to assess address utilization
- Support for MPLS VPNs with addresses assigned per subnet, per interface (see the "On-Demand Address Pools for MPLS VPNs" section on page 10-5)
- Simplified VPN setup, enabling the pool manager to request an initial subnet from the address pool server upon configuration of the on-demand address pool (ODAP)

Prerequisites for On-Demand Address Pool Manager

The on-demand address pool manager feature has the following requirements:

- You can choose to specify a VRF for an ODAP. If you do, you must configure the VRF first and then configure the VRF in the ODAP. If you do not configure a VRF in the pool, the pool is assumed to be in the global address space.
- The VRF of the PPP session must match the VRF configured in the pool. To ensure that it does, configure a virtual template interface using the **ip vrf forwarding** command. If you use AAA to authorize the PPP user, you can include the VRF in the user profile configuration on the RADIUS server.

Note For more information about configuring AAA, see the *Cisco IOS Security Configuration Guide*, Release 12.2.

Required Configuration Tasks for On-Demand Address Pool Manager

To configure the on-demand address pool manager feature, perform the following required configuration tasks:

- Defining DHCP ODAPs as the Global Default Pooling Mechanism, page 10-7
- Configuring the DHCP Pool as an ODAP, page 10-7
- Configuring the AAA Client, page 10-8
- Configuring RADIUS, page 10-9

Defining DHCP ODAPs as the Global Default Pooling Mechanism

To specify on-demand address pooling as the global default mechanism, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# ip address-pool dhcp-pool`	Enables on-demand address pooling as the global default IP address mechanism for PPP remote access sessions into MPLS VPNs. Locally configured VRF-associated DHCP pools allocate IP addresses.

Note The DHCP server needs to be able to distinguish between a normal DHCP address request and an address request for a PPP client. For more information, see the "Address Allocation for PPP Sessions" section on page 10-5.

Example 10-1 enables on-demand address pooling as the mechanism to service address requests from PPP clients. The locally configured VRF-associated DHCP pool named Green_pool provides the IP addresses.

Example 10-1 Defining DHCP ODAPs as the Global Default Pooling Mechanism

```
!
ip address-pool dhcp-pool
!
ip dhcp pool Green_pool
!
```

Configuring the DHCP Pool as an ODAP

To configure a DHCP pool as an on-demand address pool, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool` *name*	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 2	`Router(config-dhcp)# vrf` *name*	Associates the address pool with a VRF.
Step 3	`Router(config-dhcp)# origin {dhcp | aaa | ipcp} [subnet size initial` *size* `[autogrow` *size*`]]`	Configures an address pool as an on-demand address pool.
Step 4	`Router(config-dhcp)# utilizationmark low` *percentage-number*	Sets the low utilization mark of the pool size. The default value is zero percent.
Step 5	`Router(config-dhcp)# utilization mark high` *percentage-number*	Sets the high utilization mark of the pool size. The default value is 100 percent.

Example 10-2 configures two on-demand DHCP address pools: *green_pool* and *red_pool*. The *green_pool* address pool is associated with the *Green* VRF and the *red_pool* address pool is associated with the *Red* VRF. Both pools obtain their subnet addresses from an external DHCP server.

Example 10-2 Configuring the DHCP Pool as an ODAP

```
!
ip dhcp pool green_pool
    vrf Green
    utilization mark high 60
    utilization mark low 40
    origin dhcp subnet size initial /24 autogrow /24
!
ip dhcp pool red_pool
    vrf Red
    origin dhcp
!
ip vrf Green
    rd 200:1
    route-target export 200:1
    route-target import 200:1
!
ip vrf Red
    rd 300:1
    route-target export 300:1
    route-target import 300:1
ip address-pool dhcp-pool
!
interface Virtual-Template1
    ip vrf forwarding Green
    ip unnumbered Loopback1
    ppp authentication chap
!
interface Virtual-Template4
    ip vrf forwarding Red
    ip unnumbered Loopback2
    ppp authentication chap
!
```

Configuring the AAA Client

To allow an ODAP to obtain subnets from the RADIUS server, enter the following commands in global configuration mode. These commands configure the AAA client on the Cisco 10000 router:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA access control.
Step 2	`Router(config)# aaa authorization configuration default group radius`	Downloads static route configuration information from the AAA server using RADIUS.

Step 3	`Router(config)# aaa accounting network default start-stop radius`
	
	or
	
	`Router(config)# aaa accounting network default stop-only radius`
		Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a *start* accounting notice at the beginning of a process.
		
		Enables AAA accounting of requested services for billing or security purposes when you use RADIUS. Sends a *stop* accounting notice at the end of the requested user process.

Step 4	`Router(config)# aaa session-id common`	Ensures that the same session ID is used for each AAA accounting service type within a call.

For an example of how to configure AAA, see Example 10-3 in the "Configuring RADIUS" section on page 10-9.

Configuring RADIUS

To configure RADIUS on the Cisco 10000 router, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip radius source-interface subinterface-name`	Forces the Cisco 10000 router to use the IP address of the specified interface for all outgoing RADIUS packets.
Step 2	`Router(config)# radius-server host ip-address auth-port port-number acct-port port-number`	Specifies a RADIUS server host.
Step 3	`Router(config)# radius server attribute 32 include-in-access-req`	Sends RADIUS attribute 32 (NAS-Identifier) in an access request or accounting request.
Step 4	`Router(config)# radius server attribute 44 include-in-access-req`	Sends RADIUS attribute 44 (Accounting Session ID) in an access request or accounting request.
Step 5	`Router(config)# radius-server vsa send accounting`	Configures the Cisco 10000 router, acting as the network access server (NAS), to recognize and use vendor-specific accounting attributes.
Step 6	`Router(config)# radius-server vsa send authentication`	Configures the Cisco 10000 router (NAS) to recognize and use vendor-specific authentication attributes.

Example 10-3 configures an address pool named *Green* and a RADIUS server from which the *Green* address pool obtains its subnets. The RADIUS server is located at the IP address 172.16.1.1.

Example 10-3 Configuring AAA and RADIUS

```
!
aaa new-model
!
aaa authorization configuration default group radius
aaa accounting network default start-stop group radius
aaa session-id common
!
ip subnet-zero
!
ip dhcp ping packets 0
!
ip dhcp pool Green
    vrf Green
    utilization mark high 50
    utilization mark low 30
    origin aaa subnet size initial /28 autogrow /28
!
ip vrf Green
    rd 300:1
    route-target export 300:1
    route-target import 300:1
!
interface Ethernet1/1
    ip address 172.16.1.12 255.255.255.0
    duplex half
```

```
!
interface Virtual-Template1
    ip vrf forwarding Green
    no ip address
!
ip radius source-interface Ethernet1/1
!
!IP address of the Radius server host
radius-server host 172.16.1.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 32 include-in-access-req
radius-server attribute 44 include-in-access-req
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
```

Optional Configuration Tasks for On-Demand Address Pool Manager

To configure the on-demand address pool manager feature, perform any of the following optional configuration tasks:

- Defining ODAPs on an Interface, page 10-10
- Configuring ODAPs to Obtain Subnets Through IPCP Negotiation, page 10-11
- Disabling ODAPs, page 10-11

Defining ODAPs on an Interface

To configure the on-demand address pool manager feature on an interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface name`	Specifies the interface and enters interface configuration mode.
Step 2	`Router(config-if)# peer default ip address dhcp-pool`	Specifies to return an IP address from an on-demand address pool to a remote peer connecting to the interface. This command supports only remote access (PPP) sessions into MPLS VPNs.

Note When you configure the on-demand address pool mechanism on an interface-by-interface basis, the ODAP overrides the global default address pool mechanism configured on the interface.

Configuring ODAPs to Obtain Subnets Through IPCP Negotiation

Note When you assign an IP address pool to customer premise equipment (CPE), the pool manager assigns IP addresses to the CPE devices and to a DHCP pool. To use the ODAP functionality requires the following:

- The Cisco IOS CPE device must be able to request and use the subnet.
- The RADIUS server using AAA must be able to provide the subnet and insert the framed route into the proper VRF table.
- The PE router must be able to facilitate providing the subnet through IPCP.

To configure an on-demand address pool with the IP Control Protocol (IPCP) as the subnet allocation protocol, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool name`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 2	`Router(config-dhcp)# import all`	Imports option parameters into the Cisco IOS DHCP server database.
Step 3	`Router(config-dhcp)# origin ipcp`	Configures an address pool as an on-demand address pool by using IPCP as the subnet allocation protocol.
Step 4	`Router(config-dhcp)# exit`	Exits DHCP pool configuration mode.
Step 5	`Router(config)# interface type`	Selects an interface and enters interface configuration mode.
Step 6	`Router(Config-if)# ip address pool name`	Indicates to automatically configure the interface's IP address from the specified pool. **Note** The pool must be populated with a subnet from IPCP.

Disabling ODAPs

To disable an ODAP from a DHCP pool, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip dhcp pool name`	Enters DHCP pool configuration mode for the DHCP address pool indicated.
Step 2	`Router(config-if)# no origin {dhcp | aaa | ipcp}`	Disables the ODAP

Note When you disable an ODAP, all leased subnets are released. If active PPP sessions are using addresses from the released subnets, those sessions are reset. DHCP clients leasing addresses from the released subnets are not able to renew their leases.

Example 10-4 disables the on-demand DHCP pool named *test_pool*.

Example 10-4 Disabling ODAPs

```
!
ip dhcp pool test_pool
    import all
    no origin ipcp
!
```

Verifying On-Demand Address Pool Operation

To verify ODAP operation, enter the following commands in privileged EXEC mode:

Command	Purpose
`Router# show ip dhcp pool`	Displays information about all pools configured, such as high and low utilization mark, subnet size, VRF name, total addresses, and leased addresses.
`Router# show ip dhcp pool` *name*	Displays information about the specified pool, such as high and low utilization mark, subnet size, VRF name, total addresses, and leased addresses.
`Router# show ip dhcp binding`	Displays binding information for pools associated with a VRF, such as IP address, hardware address, lease expiration, and type of pool.

Example 10-5 uses the **show ip dhcp pool** command to display information for two DHCP pools: *Green* and *Global*. The *Green* pool configuration indicates:

- Autogrow—Obtain more subnets when the high-utilization mark is reached.
- Subnet size—Indicates the initial and incremental subnet sizes that the *Green* pool can request. These are the values configured using the **origin** command.
- VRF name—Indicates that the *Green* pool is associated with the *Green* VRF.
- Total addresses—Count of all the usable addresses in the pool.
- Leased addresses—Total count of the number of bindings created from the pool.
- Pending event: subnet request—Indicates that a subnet request is pending for the pool. The subnet request was scheduled because the Leased addresses count exceeds the high-utilization mark of the pool.
- Current index—Indicates the subnet address to be allocated next to the pool. In Example 10-5, three subnets are currently added. The Current index for the first two subnets is 0.0.0.0, indicating that each of these subnets has used all its available addresses.

Note The *Green* pool and the Global pool have the same 172.16.0.1 subnet allocated, which is acceptable because the *Green* pool is associated with the *Green* VRF and the Global pool is configured in the global address space.

- IP address range—Indicates the range of usable addresses from the subnet.
- Leased addresses—Indicates the individual count of bindings created from each subnet.

Example 10-5 show ip dhcp pool Command

```
Router# show ip dhcp pool

Pool Green :
    Utilization mark (high/low): 50 / 30
    Subnet size (first/next): 24 / 24 (autogrow)
    VRF name: Green
    Total addresses: 18
    Leased addresses: 13
    Pending event subnet request
    3 subnets are currently in the pool :
    Current indexIP address rangeLeased addresses
    0.0.0.0178.16.0.1- 172.16.0.66
    0.0.0.0172.16.0.9- 172.16.0.146
    172.16.0.17172.16.0.17- 172.16.0.221
Pool Global :
    Utilization mark (high/low): 100 / 0
    Subnet size (first/next): 24 / 24 (autogrow)
    Total addresses: 6
    Leased addresses: 0
    Pending event: none
    1 subnet is currently in the pool :
    Current indexIP address rangeLeased addresses
    172.16.0.1172.16.0.1- 172.16.0.60
```

Example 10-6 uses the **show ip dhcp binding** command to display the bindings from the *Green* pool. The example indicates the following:

- Type: On-demand—Indicates that the address binding is created for a PPP session.
- Lease expiration: Infinite—Indicates that the binding is valid as long as the session is up. If a subnet must be released back to the leasing server while the session is still up, the session is reset so that it is forced to obtain a new IP address.
- Hardware address—Indicates the session identifier that PPP detected for an on-demand entry.

Note Example 10-6 does not display any bindings from pools not associated with a VRF because the global pool has not allocated any addresses.

Example 10-6 show ip dhcp binding Command

```
Router# show ip dhcp binding
Bindings from all pools not associated with VRF :
IP addressHardware addressLease expirationType

Bindings from VRF pool Green :
IP addressHardware addressLease expirationType
172.16.0.15674.312d.7465.7374.InfiniteOn-demand
    2d38.3930.39
172.16.0.25674.312d.7465.7374.InfiniteOn-demand
    2d38.3839.31
172.16.0.35674.312d.7465.7374.InfiniteOn-demand
    2d36.3432.34
172.16.0.45674.312d.7465.7374.InfiniteOn-demand
    2d38.3236.34
172.16.0.55674.312d.7465.7374.InfiniteOn-demand
    2d34.3331.37
172.16.0.65674.312d.7465.7374.InfiniteOn-demand
    2d37.3237.39
172.16.0.95674.312d.7465.7374.InfiniteOn-demand
    2d39.3732.36
172.16.0.105674.312d.7465.7374.InfiniteOn-demand
    2d31.3637
172.16.0.115674.312d.7465.7374.InfiniteOn-demand
    2d39.3137.36
172.16.0.125674.312d.7465.7374.InfiniteOn-demand
    2d37.3838.30
172.16.0.135674.312d.7465.7374.InfiniteOn-demand
    2d32.3339.37
172.16.0.145674.312d.7465.7374.InfiniteOn-demand
    2d31.3038.31
172.16.0.175674.312d.7465.7374.InfiniteOn-demand
    2d38.3832.38
172.16.0.185674.312d.7465.7374.InfiniteOn-demand
    2d32.3736.31
```

Configuration Examples for On-Demand Address Pool Manager

This section provides the following configuration examples:

- Configuring DHCP ODAPs on an Interface, page 10-14
- Configuring ODAPs to Obtain Subnets Through IPCP Negotiation, page 10-15

Configuring DHCP ODAPs on an Interface

Example 10-7 defines ODAPs on a virtual template interface named *Virtual-Template1*. Remote peers connecting to an interface on which *Virtual-Template1* is applied obtain their IP addresses from the ODAP.

Example 10-7 Defining DHCP ODAPs on an Interface

```
!
interface Virtual-Template1
    ip vrf forwarding green
    ip unnumbered loopback1
    ppp authentication chap
    peer default ip address dhcp-pool
```

Configuring ODAPs to Obtain Subnets Through IPCP Negotiation

Example 10-8 creates a DHCP address pool named *my_pool*, configures the pool as an on-demand address pool using IPCP as the subnet allocation protocol, and configures the Ethernet0 interface to automatically obtain its IP address from the *my_pool* address pool.

Example 10-8 Enabling ODAPs to Obtain Subnets Through IPCP Negotiation

```
!
ip dhcp pool my_pool
    import all
    origin ipcp
!
interface Ethernet0
    ip address pool my_pool
    ip verify unicast reverse-path
    shutdown
    hold-queue 32 in
!
```

Monitoring and Maintaining an On-Demand Address Pool

To monitor and maintain an ODAP, enter the following commands in privileged EXEC mode:

Command	Purpose
Router# **clear ip dhcp** [**pool** *name*] **binding** {***** | *address*}	Deletes an automatic address binding or objects for a specific pool from the DHCP server database.
Router# **clear ip dhcp** [**pool** *name*] **conflict** {***** | *address*}	Clears an address conflict(s) for a specific pool from the DHCP server database.
Router# **clear ip dhcp** [**pool** *name*] **subnet** {***** | *address*}	Clears all currently leased subnets in the specified DHCP pool or in all DHCP pools if you do not specify a specific pool.
Router# **debug dhcp details**	Monitors subnet allocation and subnet releases for the on-demand address pools.
Router# **debug ip dhcp server events**	Reports DHCP server events, such as assignments and database updates.
Router# **show ip dhcp import**	Displays the option parameters imported into the DHCP server database.

Command	Purpose
`Router# show ip interface [type number]`	Displays the usability status of interfaces configured for IP.
`Router# show ip dhcp pool name`	Displays DHCP address pool information. Use this command to check that the DHCP pool assigns an IP address for each incoming PPP session and associates the address with the correct VRF.

Tip

- By default, the Cisco IOS DHCP server that the pool manager uses verifies address availability by using the **ping** command before allocating the address; the default DHCP ping configuration waits two seconds for an ICMP echo reply. As a result of this default configuration, the DHCP server services one address request every two seconds. You can configure the number of ping packets sent and the ping timeout timer. To reduce the address allocation time, reduce either the timeout timer value or the number of ping packets sent.

  Note While reducing the address allocation time improves address allocation, the reduced time inhibits the DHCP server's ability to detect duplicate addresses.

- Each ODAP retries up to four times to obtain a subnet from the DHCP server or the RADIUS server. If unsuccessful, the subnet request automatically starts when another individual address request is made to the pool (for example, a newly brought up PPP session makes an address request). If the address allocation server has not allocated any subnets to a pool, you can force the subnet request process to restart by using the **clear ip dhcp pool name subnet *** command in privileged EXEC mode.

Overlapping IP Address Pools

The Overlapping IP Address Pools feature enables you to use multiple IP address spaces and reuse IP addresses among different VPNs supported on the Cisco 10000 router. Duplicate IP addresses cannot reside in the same IP address space.

To uniquely place IP addresses within a given IP address space, multiple address spaces are assigned to IP address groups. This also allows for the verification of nonoverlapping IP address pools within an IP address group. Within the Cisco 10000 router, use unique pool names. Each pool name has an implicit group identifier to ensure that it is associated with only one group.

The Cisco 10000 router considers pools without an explicit group name as members of a base system group and processes these pools as if the IP addresses belong to a single IP address space. You cannot assign a given IP address multiple times from the pool of a single IP address space.

Existing configurations are not affected by the Overlapping IP Address Pools feature. The processing of pools that are not specified as a member of a group is unchanged from the existing implementation.

The Overlapping IP Address Pools feature is useful in the following deployment models:

- Managed L2TP Network Server
- PPP Terminated Aggregation (PTA) to VRF
- Remote Access (RA) to MPLS VPN

The Overlapping IP Address Pools feature is described in the following topics:

Feature History for Overlapping IP Address Pools

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7)XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for Overlapping IP Address Pools

The software checks for duplicate addresses on a per-group basis. This means that you can configure pools in multiple groups that could have possible duplicate addresses. You should only use this feature in environments such as MPLS VPN where multiple IP address spaces are supported.

Configuration Tasks for Overlapping IP Address Pools

To configure the IP overlapping address pools feature, configure a local pool group as described in "Configuring a Local Pool Group for IP Overlapping Address Pools".

Configuring a Local Pool Group for IP Overlapping Address Pools

To configure a local pool group, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# ip local pool pool-name start-IP [end-IP] [group group-name] [cache-size size]`	Configures a group of local IP address pools, gives the group a name, and specifies a cache size.

Verifying Local Pool Groups for IP Overlapping Address Pools

To verify that you have successfully configured a pool group, enter the following commands in privileged EXEC mode and check the resulting output for the pool group name:

Command	Purpose
`Router# show ip local pool [pool-name [group group-name]]`	Displays statistics for defined IP address pools.
`Router# show ip local pool`	Displays statistics for all pools configured.
`Router# show ip local pool pool-name`	Displays statistics for a specific pool you specify.
`Router# show ip local pool group`	Displays statistics for all pools in a base system group.
`Router# show ip local pool group group-name`	Displays all pools in a specified group.

Configuration Examples for Overlapping IP Address Pools

This section provides the following configuration examples:

- Generic IP Overlapping Address Pools Example, page 10-18
- IP Overlapping Address Pools for VPNs and VRFs Example, page 10-19

Generic IP Overlapping Address Pools Example

The following example shows the configuration of two pool groups and includes pools in the base system group. In this example:

- Pool group grp1 consists of pools p1_g1, p2_g1, and p3_g1.
- Pool group grp2 consists of pools p1_g2 and p2_g2.
- Pools lp1 and lp2 are members of the base system group.
- The IP address 10.1.1.1 overlaps grp1, grp2, and the base system group.
- No overlapping addresses occur within any group including the unnamed base system group, which consists of pools lp1 and lp2.

```
ip local pool p1_g1 10.1.1.1 10.1.1.50 group grp1
ip local pool p2_g1 10.1.1.100 10.1.1.110 group grp1
ip local pool p1_g2 10.1.1.1 10.1.1.40 group grp2
ip local pool lp1 10.1.1.1 10.1.1.10
ip local pool p3_g1 10.1.2.1 10.1.2.30 group grp1
ip local pool p2_g2 10.1.1.50 10.1.1.70 group grp2
ip local pool lp2 10.1.2.1 10.1.2.10
```

Note The preceding example shows pool names that provide an easy way to associate a pool name with a group (when the pool name stands alone). This association is an operational convenience. There is no required relationship between the names used to define a pool and the name of the group.

IP Overlapping Address Pools for VPNs and VRFs Example

The following example is a general IP address configuration that VPNs and VRFs might use. This example shows pool names that provide a way to associate a pool name with a VPN (when the pool name stands alone). This association is an operational convenience. There is no required relationship between the names used to define a pool and the name of the group. In this example:

- Pool group vpn1 consists of pools p1_vpn1, p2_vpn1, and p3_vpn1.
- Pool group vpn2 consists of pools p1_vpn2, p2_vpn2.
- Pools lp1 and lp2 are members of the base system.
- The IP address 10.1.1.1 overlaps vpn1, vpn2, and the base system group.
- No overlapping addresses occur within any group including the unnamed base system group, which consists of pools lp1 and lp1.

```
ip local pool p1_vpn1 10.1.1.1 10.1.1.50 group vpn1
ip local pool p2_vpn1 10.1.1.100 10.1.1.110 group vpn1
ip local pool p1_vpn2 10.1.1.1 10.1.1.40 group vpn2
ip local pool lp1 10.1.1.1 10.1.1.10
ip local pool p3_vpn1 10.1.2.1 10.1.2.30 group vpn1
ip local pool p2_vpn2 10.1.1.50 10.1.1.70 group vpn2
ip local pool lp2 10.1.2.1 10.1.2.10
```

CHAPTER 11

Configuring Local AAA Server, User Database—Domain to VRF

The Local AAA Server, User Database—Domain to VRF feature extends the Cisco IOS AAA Authorization to local AAA profiles on the router without using an AAA Server. The local user database acts as a local AAA server, and is fully compatible with any external AAA Server. If you want to maintain your user database locally or provide a failover local mechanism, you no longer have to sacrifice policy options when defining local users.

This flexibility allows you to provide complete user authentication and authorization locally within Cisco IOS without using an AAA Server, provided the local username list is relatively small. While authentication can be done on the router for a limited number of user names, it might make more sense and be much more scalable to use an AAA Server. Note that accounting is still be done on an AAA server and is not be supported on the router.

The key function that this feature provides is a mapping of user domain names to local AAA profiles. This allows AAA attributes to be applied to the PPP session as part of the PPP session establishment. These local AAA attributes are RADIUS attributes that would normally be defined on a Radius Server but now are defined locally on the router.

Subscriber profiles are used to match user domain names, and on a match to use a defined AAA attribute list. The AAA attribute list contains a list of valid Cisco IOS format AAA attributes.

Note Domain to subscriber profile matching is a global match. Limiting which domains are permitted or denied per PPPoE bba-group or PVC is not supported.

This chapter describes the Local AAA Server, User Database—Domain to VRF feature in the following topics:

Feature History for Local AAA Server, User Database—Domain to VRF

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Prerequisites for Local AAA Server, User Database—Domain to VRF

The Local AAA Server, User Database—Domain to VRF feature has the following requirements:

- Configure an external AAA as described in *Cisco IOS Security Configuration Guide, Cisco IOS Release 12.2*.

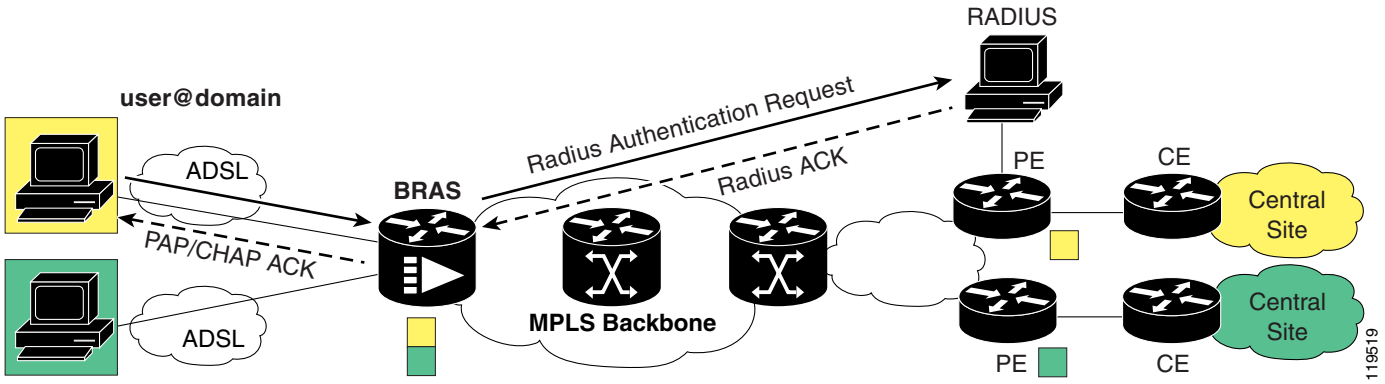
Establishing a PPP Connection

The following example describes the sequence of events involved in setting up AAA authentication, authorization, and accounting when a PPP connection is established and a local AAA server is used.

AAA Authentication

Figure 11-1 shows the AAA authentication set up when establishing a PPP connection.

Figure 11-1 AAA Authentication

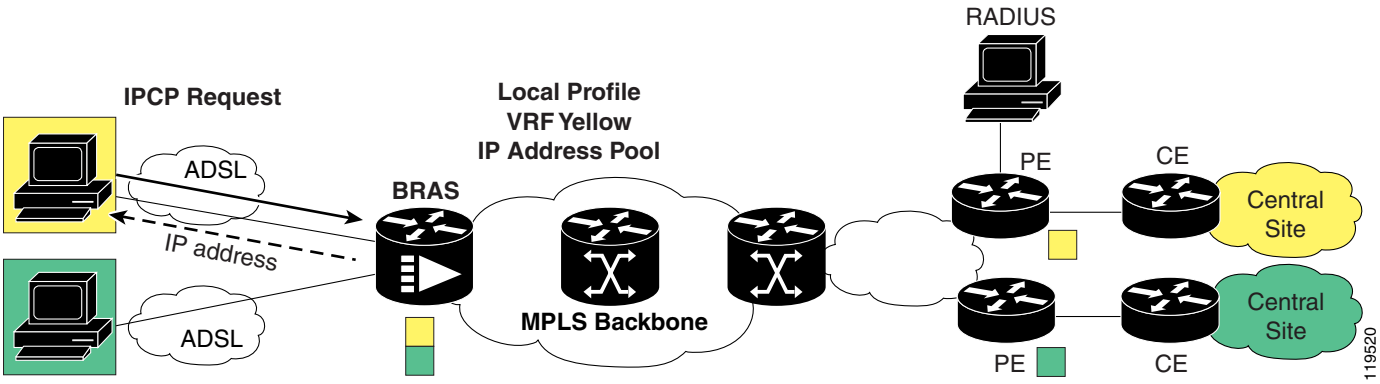


In the figure, the PPP client attempts to establish a PPP session with user@domain. This PAP or CHAP user name request is forwarded to the broadband remote access server (BRAS) for authentication. Authentication could be done locally on the BRAS, but in most cases the authentication is forwarded to a RADIUS server. The RADIUS server looks up the user@domain or user (if the BRAS strips off the domain), and if found sends a RADIUS ACK back to the BRAS. The BRAS sends a PAP or CHAP ACK back to the PPP client.

AAA Authorization

Figure 11-2 shows the AAA authorization set up when establishing a PPP connection.

Figure 11-2 AAA Authorization

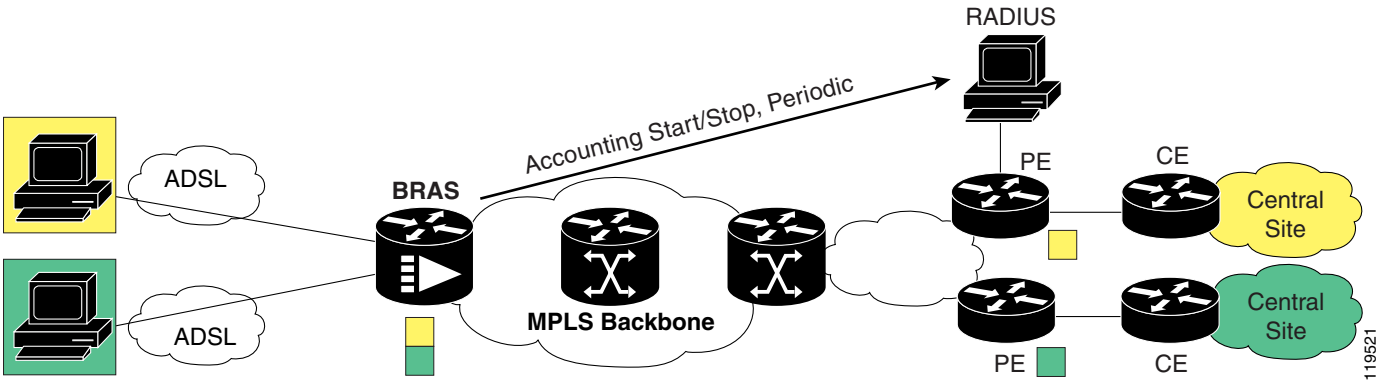


In the figure, the PPP client requests an IP address using PPP IPCP to the BRAS. The BRAS does a match of the domain to a local profile. This local profile contains the VRF to assign to this PPP session. The BRAS replies back to the PPP client with an IP address from the defined IP address pool in the local profile.

AAA Accounting

Figure 11-3 shows the AAA accounting set up when establishing a PPP connection.

Figure 11-3 AAA Accounting



In the figure, the BRAS can be configured to provide AAA accounting start/stop and periodic records for each PPP session. The BRAS can also be configured to provide NAS-Port information in the accounting records that will detail the slot/card/interface and VPI/VCI or VLAN.

AAA Attribute Lists

AAA Attribute Lists are used by the subscriber profiles when there is a match of the user name domain. These lists define RADIUS user profiles local to the router. The attributes are available for configuration using the **aaa attribute list** *name* global configuration command. Every attribute known to AAA is made available for configuration.

The AAA attributes that are defined in the AAA attribute list are standard RADIUS attributes, however they are in the Cisco IOS AAA format of the attribute. You must convert the attributes from RADIUS format to Cisco IOS AAA format.

Converting from RADIUS Format to Cisco IOS AAA Format

Use the **show aaa attribute protocol radius** command to get the Cisco IOS AAA format of the IETF RADIUS Attribute. This provides a complete list of all the aaa attributes supported. The following is an example where you need to convert the RADIUS attribute Filter-Id to Cisco IOS AAA format. This example represents part of the output of the **show aaa attribute protocol radius** command.

IIETF defined attributes:

```
Type=4     Name=acl                              Format=Ulong
Protocol:RADIUS
Unknown       Type=11    Name=Filter-Id             Format=Binary
```

Cisco IOS converts the IETF RADIUS attribute 11 (Filter-Id) of type Binary into an internal attribute named acl of type Ulong. Now you can configure this attribute locally using the attribute type acl.

Note You cannot add new AAA attributes during the conversion process. The conversion is only making the attributes configurable and usable locally on the router. The defined local AAA attributes must be supported RADIUS attributes.

Defining AAA Attribute Lists

Typically, you define an AAA attribute list for each user name domain. Cisco IOS Release 12.3(7)XI1 introduces the following two new commands to define local AAA attribute lists and attribute types:

Command	Purpose
Router(config)# **aaa attribute list** *aaa attribute list name*	Defines an AAA attribute list locally on the router. This attribute list is applied to the PPP session. *aaa attribute name* is the name of the local AAA attribute list.
Router(config)# **aaa attribute type** *name value* [**service ppp**] [**protocol** {**ip** | **atm** | **vpdn**}] [*tag*]	Defines an AAA attribute locally on the router. These attributes are RADIUS attributes in Cisco IOS AAA format. *name* defines the Cisco IOS AAA internal name of the IETF RADIUS attribute. *value* defines a string, binary, or IPv4 address value. This is the RADIUS attribute that is being defined but in IOS AAA format. **service** defines the access method, which is typically PPP. **protocol** can be ip, atm, or vpdn. *tag* provides a means of grouping attributes that refer to the same VPDN tunnel.

The following is an example of the commands you use to configure method lists:

```
aaa attribute list <name>
attribute type <name> <value> <service> <protocol> <tag>
```

Subscriber Profiles

Subscriber profiles are used to match user domain names, and on a match to use a defined AAA attribute list. Cisco IOS Release 12.3(7)XI1 introduces the following new command to define subscriber profiles:

Command	Purpose
Router(config)# **subscriber profile** *domain-name*	Defines an AAA attribute list locally on the router. This attribute list is applied to the PPP session. *domain-name* is the PPP user name domain.

The following is an example of the commands you use to configure a subscriber profile:

```
subscriber authorization enable
subscriber profile domain-name
service local
aaa attribute list aaa attribute list name
```

AAA Method Lists

The AAA method lists are defined to use RADIUS for authentication and accounting. Authorization is done locally using the AAA attribute lists. Defining the AAA attribute lists for PPP under the virtual template no longer requires defining the AAA lists. Instead, a default authentication and authorization list can be defined on the virtual template and the AAA method lists can be defined in the AAA attribute lists. 2000 method lists are supported.

Using method lists does require that you define **aaa authentication ppp default** and **aaa authorization network default** lists. The following is an example of the commands you use to configure method lists:

```
interface virtual-template
ppp authentication pap chap

aaa new-model
aaa authentication ppp default local
aaa authorization network default local
aaa authentication ppp method list name group radius
aaa authorization network method list name local if-authenticated
aaa accounting network method list name start-stop group radius

aaa attribute list <domain name>
attribute type ppp-authen-list "method list name"
attribute type ppp-author-list "method list name"
attribute type ppp-acct-list "method list name"
```

Configuration Tasks for Local AAA Server, User Database—Domain to VRF Using Local Attributes

To configure a user name domain to a VRF using local AAA attributes, perform the following configuration tasks:

- Defining AAA, page 11-6
- Defining RADIUS and Enabling NAS-PORT, page 11-7
- Defining a VRF, page 11-7
- Applying AAA to a Virtual Template, page 11-7
- Defining a Loopback Interface, page 11-8
- Creating an IP Address Pool, page 11-8
- Defining a Subscriber Profile, page 11-8
- Defining an AAA Attribute List, page 11-8

Defining AAA

To define AAA (authentication, authorization, and accounting), enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router(config)# aaa authentication ppp` *list-name* `group radius`	Specifies RADIUS to authenticate the PPP user name.

	Command	Purpose
Step 3	`Router(config)# aaa authorization network list-name local if-authenticated`	Specifies to use the local profile if authenticated.
Step 4	`Router(config)# aaa accounting network list-name start-stop group radius`	Specifies RADIUS accounting as optional.
Step 5	`Router(config)# aaa authentication ppp default local`	Required to allow the definition of the AAA authentication list in the AAA attribute list.
Step 6	`Router(config)# aaa authorization network default local`	Required to allow the definition of the AAA authorization list in the AAA attribute list.

Defining RADIUS and Enabling NAS-PORT

To define RADIUS and enable NAS-PORT, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host ip-address auth-port 1645 acct-port 1646 key password`	Defines the Radius server that AAA authentication, authorization and accounting requests are sent to.
Step 2	`Router(config)# radius-server attribute nas-port format d`	Defines NAS-Port information to be sent to the AAA accounting server. (optional)

Defining a VRF

To define a VRF, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip vrf vrf-name`	Enters VRF configuration mode and defines the VRF instance by assigning a VRF name.
Step 2	`Router(config-vrf)# rd route-distinguisher`	Creates routing and forwarding tables.
Step 3	`Router(config-vrf)# route-target {import | export | both} route-target-ext-community`	Creates a list of import an export route target communities for the specified VRF.

Applying AAA to a Virtual Template

To apply AAA to a virtual template, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Defines the virtual template to use for PPP.
Step 2	`Router(config)# ppp mtu adaptive`	For PPPoE, defines auto negotiation of MTU size.
Step 3	`Router(config)# ppp authentication pap chap`	Enables PAP, then CHAP, for PPP authentication.

Defining a Loopback Interface

To define a loopback interface, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface loopback number`	Defines a loopback for the PPP session.
Step 2	`Router(config)# ip vrf forwarding vrf name`	Enables VRF forwarding.
Step 3	`Router(config)# ip address address mask`	Sets the IP address.

Creating an IP Address Pool

To an IP address pool, enter the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip local pool start address end address`	Defines an IP pool from which the PPP sessions are IP addresses.

Defining a Subscriber Profile

To define a subscriber profile, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# subscriber authorization enable`	Enables subscriber authorization.
Step 2	`Router(config)# subscriber profile domain-name`	Specifies the user name domain to match.
Step 3	`Router(config)# service local`	Specifies to perform local subscriber authorization.
Step 4	`Router(config)# aaa attribute list aaa attribute-list name`	Defines the AAA attribute list from which to get RADIUS attributes and that is applied to the PPP session.

Defining an AAA Attribute List

To define AAA attribute list, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa attribute list aaa attribute-list name`	Defines an AAA attribute list.
Step 2	`Router(config)# attribute type addr-pool pool_name protocol ip`	Defines an IP address pool to use.
Step 3	`Router(config)# attribute type ip-unnumbered loopback number service ppp protocol ip`	Defines the loopback interface to use.
Step 4	`Router(config)# attribute type vrf-id vrf_name service ppp protocol ip`	Defines the VRF to use.

	Command	Purpose
Step 5	`Router(config)# attribute type ppp-authen-list aaa_list_name`	Defines the AAA authentication list to use.
Step 6	`Router(config)# attribute type ppp-author-list aaa_list_name`	Defines the AAA authorization list to use.
Step 7	`Router(config)# attribute type ppp-acct-list aaa_list_name`	Defines the AAA accounting list to use.

Verifying Local AAA Server, User Database—Domain to VRF Using Local Attributes

To verify domain to VRF using local attributes, use the **show aaa users all** command and the **show running-config** command. See the next section for a configuration example.

Configuration Example for Local AAA Server, User Database—Domain to VRF

The following configuration example has two subscriber profiles that match on domain cisco1.com and cisco2.com.

A subscriber with the domain name cisco1.com uses the parameters defined in the subscriber profile cisco1.com. The name of the subscriber profile must be identical to the domain part of the full username (username@domain). An attribute list cisco1.com defined in the service profile is used to reference AAA attributes for the PPP subscribers.

Subscriber cisco1.com is applied with AAA attributes from AAA attribute list cisco1.com. An attribute is applied to put the PPP session into a VRF called vrf1. An IP address is assigned from a local DHCP pool called dhcp-pool. AAA authentication, authorization, and accounting are also defined and use an AAA list called test1. These all use an AAA group server called group_server_test1.

A subscriber with the domain name cisco2.com uses the parameters defined in the subscriber profile cisco2.com. The name of the subscriber profile must be identical to the domain part of the full username (username@domain). An attribute list cisco2.com defined in the service profile is used to reference aaa attributes for the PPP subscribers.

Subscriber cisco2.com is applied with AAA attributes from AAA attribute list cisco2.com. An attribute is applied to put the PPP session into a VRF called vrf2. An IP address is assigned from a local pool called pppoe2. AAA authentication, authorization, and accounting are also defined and use an AAA list called test2. These all use an AAA group server called group_server_test2.

```
aaa new-model
!
!
aaa group server radius group_server_test1
 server-private 192.168.2.20 auth-port 1645 acct-port 1646 key cisco
 ip vrf forwarding vrf1
!
aaa group server radius group_server_test2
 server-private 192.168.2.12 auth-port 1645 acct-port 1646 key cisco
 ip vrf forwarding vrf2
!
aaa authentication ppp default local
aaa authentication ppp test1 group test1
aaa authentication ppp test2 group test2
aaa authorization network default local
aaa authorization network test1 local if-authenticated
```

```
aaa authorization network test2 local if-authenticated
aaa accounting delay-start all
aaa accounting network test1 start-stop group group_server_test1
aaa accounting network test2 start-stop group group_server_test2
!
aaa attribute list cisco1.com
 attribute type addr-pool "dhcp-pool" protocol ip
 attribute type ip-unnumbered "loopback1" service ppp protocol ip
 attribute type vrf-id "vrf1" service ppp protocol ip
 attribute type ppp-authen-list "test1"
 attribute type ppp-author-list "test1"
 attribute type ppp-acct-list "test1"
!
aaa attribute list cisco2.com
 attribute type addr-pool "pppoe2" protocol ip
 attribute type ip-unnumbered "loopback2" service ppp protocol ip
 attribute type vrf-id "vrf2" service ppp protocol ip
 attribute type ppp-authen-list "test2"
 attribute type ppp-author-list "test2"
 attribute type ppp-acct-list "test2"
!
ip dhcp pool dhcp-pool
   vrf vrf1
   network 101.1.0.0 255.255.0.0
   default-router 100.1.1.1
   lease 0 2 30
!
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
ip vrf vrf2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
!
subscriber authorization enable
!
subscriber profile cisco1.com
 service local
 aaa attribute list cisco1.com
!
subscriber profile cisco2.com
 aaa attribute list cisco2.com
!
vpdn enable
!
ppp hold-queue 80000
no virtual-template snmp
!
buffers small permanent 15000
buffers middle permanent 12000
buffers large permanent 1000
!
bba-group pppoe cisco1.com
 virtual-template 1
!
bba-group pppoe cisco2.com
 virtual-template 2
!
interface Loopback1
 ip vrf forwarding vrf1
 ip address 100.1.1.1 255.255.255.255
```

```
!
interface Loopback2
 ip vrf forwarding vrf2
 ip address 101.1.1.1 255.255.255.255
!
interface FastEthernet0/0/0
 shutdown
!
interface ATM1/0/0
 no ip address
 no atm pxf queuing
 no atm ilmi-keepalive
!
interface ATM1/0/0.1 multipoint
 pvc 1/32
  encapsulation aal5autoppp Virtual-Template1 group cisco1.com
  no create on-demand
!
!
interface ATM1/0/0.2 multipoint
 pvc 1/33
  encapsulation aal5autoppp Virtual-Template2 group cisco2.com
!
!
interface FastEthernet6/0/0
 ip vrf forwarding vrf1
 ip address 192.168.2.201 255.255.255.0
 duplex auto
!
interface FastEthernet6/0/1
 ip vrf forwarding vrf2
 ip address 192.168.2.202 255.255.255.0
 duplex auto
!
interface Virtual-Template1
 no ip address
 no logging event link-status
 no snmp trap link-status
 ppp mtu adaptive
 ppp authentication chap callin
!
ip local pool pppoe2 12.1.1.1 12.1.250.1
!
ip radius source-interface FastEthernet6/0/0.1 vrf vrf1
ip radius source-interface FastEthernet6/0/0.2 vrf vrf2
!
radius-server attribute nas-port format d
radius-server domain-stripping
```

Example—VRF with DBS

Applying the PCR and SCR to this PPP:

```
aaa attribute list cisco1.com
attribute type addr-pool "pppoe" protocol ip
attribute type ip-unnumbered "loopback1" service ppp protocol ip
attribute type vrf-id "vrf1" service ppp protocol ip
attribute type peak-cell-rate 2048 protocol atm
attribute type sustainable-cell-rate 1024 protocol atm
```

Example—VRF with ACL

Applying a defined output ACL to this PPP:

```
aaa attribute list cisco1.com
attribute type addr-pool "pppoe" protocol ip
attribute type ip-unnumbered "loopback1" service ppp protocol ip
attribute type vrf-id "vrf1" service ppp protocol ip
attribute type outacl "101" service ppp protocol ip

access-list 101 deny   icmp any any
```

Monitoring and Maintaining Local AAA Server, User Database—Domain to VRF

The following debug commands can be helpful in monitoring and maintaining Local AAA Server, User Database—Domain to VRF:

- **debug aaa id**—displays a unique key for a session and provides a way to track sessions
- **debug aaa authentication**—displays the methods of authentication being used and the results of these methods
- **debug aaa authorization**—displays the methods of authorization being used and the results of these methods
- **debug aaa per-user**—displays information about per-user QoS parameters
- **debug ppp negotiation**—shows PPP negotiation debug messages
- **debug ppp authen**—indicates if a client is passing authentication
- **debug ppp error**—displays protocol errors and error statistics associated with PPP connection negotiation and operation
- **debug ppp forward**—displays who is taking control of a session
- **debug sss error**—displays diagnostic information about errors that may occur during Subscriber Service Switch (SSS) call setup
- **debug radius**—displays information about the RADIUS server

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

CHAPTER 12

Configuring Traffic Filtering

The Cisco 10000 series router provides traffic filtering capabilities using access control lists (ACLs). Access lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. Using ACLs, you can do such things as restrict the contents of routing updates, provide traffic flow control, and provide security for your network.

The Cisco 10000 series router supports the following ACL types and features:

- Standard and extended ACLs
- Named and numbered ACLs
- Turbo-ACLs
- Per-user ACLs
- IP receive ACLs
- Time-based ACLs

For more information about ACLs, see the following documents:

- *Turbo Access Control Lists, Release 12.1(5)T* feature module
- Part 3: Traffic Filtering and Firewalls in the *Cisco IOS Security Configuration Guide, Release 12.2*

This chapter describes the following features:

IP Receive ACLs

The IP Receive ACLs feature provides basic filtering capability for traffic that is destined for the router and protects the router from remote intrusions.

To restrict access to the router, you apply a numbered ACL to the ingress interface of the router. You can restrict access to the router to known and trusted sources, and to expected traffic profiles. The IP Receive ACLs feature supports both standard and extended ACLs. The rules for numbered ACLs also apply to the access control entries (ACEs) of the IP receive ACL.

The IP receive ACL filters traffic on the parallel express forwarding engine (PXF) before filtering the packets received by the route processor (RP). This feature protects the router from denial of service (DoS) floods, thereby preventing the flood from degrading the performance of the route processor (RP).

The IP Receive ACLs feature is described in the following topics:

Feature History for IP Receive ACLs

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2

Restrictions for IP Receive ACLs

The IP receive ACLs feature has the following restrictions:

- A receive ACL must be a numbered ACL. You cannot use a named ACL as the receive ACL.
- The rules for numbered ACLs also apply to the access control entries (ACEs) of receive ACLs.
- Time-based and reflexive ACLs are not supported as receive ACLs.
- Only traffic processed by the RP is filtered. Traffic that is processed exclusively by the Forwarding Processor (FP) is not filtered. For example, GRE tunneled packets, L2TP tunneled packets, and some ICMP packets are not filtered.

Configuration Tasks for IP Receive ACLs

To configure the IP Receive ACLs feature, perform the following configuration tasks:

Configuring Receive ACLs

To configure receive ACLs, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip receive acl number`	Activates receive ACLs and begins filtering packets destined for the router.

Step 2	`Router(config)# access-list access-list-number {deny | permit} source [source-wildcard] [log]`
	
	`or`
	
	`Router (config)# access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]`
		Defines a standard IP access list.
		
		Defines an extended IP access list.
		
		Note The **timeout** argument and the **time-range** argument are not supported on Cisco IOS Release 12.3(7)XI1.

Verifying Receive ACLs

To verify the configuration of receive ACLs, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show access-lists`	Displays the contents of all current standard and extended access lists. (Default)
`Router# show access-lists [access-list-number | access-list-name]`	Displays the contents of the access list you specify.
`Router# show ip access-list`	Displays the contents of all current standard and extended IP access lists. (Default)
`Router# show ip access-list [access-list-number | access-list-name]`	Displays the contents of the IP access list you specify.

Configuration Example for IP Receive ACLs

Example 12-1 shows how to configure an extended IP receive ACL. The ACEs of this numbered ACL (100) do the following:

- Deny fragmented ping operations
- Permit the router to respond to ping operations
- Permit FTP operations from network 192.168.1.0
- Permit OSPF routing updates
- Permit BGP routing updates from the host 10.0.0.1
- Deny any other IP traffic

Example 12-1 Receive ACL Configuration

```
ip receive access-list 100
access-list 100 deny icmp any any fragments
access-list 100 permit icmp any any echo
access-list 100 permit tcp 192.168.1.0 0.0.0.255 any eq 22
access-list 100 permit ospf any any precedence internet
access-list 100 permit tcp host 10.0.0.1 any eq bgp precedence internet
access-list 100 deny ip any any
```

Time-Based ACLs

The Time-based ACLs feature allows the network administrator to define a time range when certain resources may be accessed, thus providing greater control over resource usage.

While functionally similar to extended ACLs, time-based ACLs control access to the router for a specific time period. A time range, identified by a name, defines the specific times of the day and week that the ACL is active. The access control entries (ACEs) reference the time range name, which imposes the time restriction on the ACEs. The time range relies on router's system clock to activate or deactivate an ACE.

Previously, access list statements were always in effect after they were applied to an interface. However, using the time-range command, network administrators can now define when the permit and deny statements in the ACL are in effect. Both named and numbered access lists can reference a time range.

When you create a time range, you can specify both absolute and periodic time entries. The **periodic** command in time-range configuration mode allows you to specify the days of the week and the time of day that the access control entry (ACE) is active. The **absolute** command in time-range configuration mode allows you to specify a specific time and date to activate the ACE and a specific time and date to stop processing the ACE. You can specify only one absolute entry for each time range. During ACL processing, the router begins evaluating the time range entry attached to the ACE after it reaches the absolute start time. The router then evaluates the periodic values until the router reaches the absolute end entry. No further processing occurs after the router reaches the absolute end value.

The Tine-based ACLs feature is described in the following topics:

Feature History for Time-Based ACLs

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for Time-Based ACLs

The Time-Based ACLs feature has the following restrictions:

- You can specify a time range for only IP extended access lists. Standard access lists are not supported.
- An ACE that refers to a non-existent time-range entry is considered active.
- You define time-based ACLs based on hours and minutes. You cannot specify seconds.

Configuration Tasks for Time-Based ACLs

To configure the Time-Based ACLs feature, perform the following configuration tasks:

- Creating a Time Range, page 12-5
- Applying a Time Range to a Numbered Access Control List, page 12-6
- Applying a Time Range to a Named Access Control List, page 12-7

Creating a Time Range

To create a time range, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# time-range name`	Defines a named time range and enters time-range configuration mode.
Step 2	`Router(config-time-range)# periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm`	(Optional) Defines the periodic times that the time range is active. Valid values for *days-of-the-week* are **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, and **Sunday**. You can also specify **daily** for Monday through Sunday, **weekdays** for Monday through Friday, and **weekend** for Saturday and Sunday. The *hh:mm* argument specifies hours:minutes in a 24 hour format. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m. The ending *days-of-the-week* argument defaults to the value you specify in the beginning *days-of-the-week* argument. Specify the ending *days-of-the-week* only if it is different from the beginning *days-of-the-week*.
Step 3	`Router(config-time-range)# absolute [start time date] [end time date]`	(Optional) Defines the absolute times that the time range is active. You specify the **start** and **end** arguments in the format of *hh:mm day month year*, using a 24 hour format. The minimum **start** value is **00:00 1 January 1993**. If you do not specify a start value, it defaults to **right now.** The maximum **end** value is **23:59 31 December 2035**. The **end** value must be greater than the **start** value; otherwise, an error occurs. If you do not specify an **end** value, it defaults to **forever after the starting time**. **Note** You can specify only one absolute entry for each time range you create.

Example 12-2 creates a periodic time range named *no-http* that specifies Monday through Friday from 8:00 a.m. to 6:00 p.m.

Example 12-2 Configuring a Time Range

```
Router(config)# time-range no-http
Router(config-time-range)# periodic weekdays 8:00 to 18:00
```

Example 12-3 creates a time range named *HTTP* that specifies both periodic and absolute values. During ACL processing, the router assumes that the time period begins right now because the **absolute** command does not specify a **start** value. The router then evaluates the **periodic** value, which indicates that the time period is restricted to Monday through Wednesday from 8:00 a.m. to 7:00 p.m. The time period ends on February 6 at 11:59 p.m.

Example 12-3 Configuring a Time Range with Periodic and Absolute Entries

```
Router(config)# time-range http
Router(config-t-range)# periodic monday 8:00 to wednesday 19:00
Router(config-t-range)# absolute end 23:59 6 February 2000
```

Applying a Time Range to a Numbered Access Control List

To apply a time range to the access control entries (ACEs) of a numbered extended access control list (ACL), enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# access-list` *access-list-number* `[dynamic` *dynamic-name* `[timeout` *minutes*`]] {deny | permit}` *protocol source source-wildcard destination destination-wildcard* `[precedence` *precedence*`] [tos` *tos*`] [log | log-input] time-range` *time-range-name* `[fragments]`	Defines a numbered extended IP access control list (ACL). The **time-range** *time-range-name* argument specifies the name of the time range to apply to the ACE. **Note** In Cisco IOS Release 12.3(7)XI1, the **time-range** argument is required. For more information about the access-list command, see the *Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services, Release 12.3*.
Step 2	`Router(config)# interface` *type number slot/module/port.subinterface*	Configures an interface and enters interface configuration mode.
Step 3	`Router(config-if)# ip access-group {`*access-list-number* `|` *access-list-name*`} {in | out}`	Controls access to an interface.

Example 12-4 permits SMTP traffic to the access the mail host (128.88.1.2) on Monday through Sunday between the hours of 5:00 a.m. and 11:59 p.m, if the traffic belongs to an already established connection. The example creates the time range named *smtp* and applies it to the ACE of the extended access list numbered 102. The time-based ACL is then applied to the ingress serial 0 interface.

Example 12-4 Applying a Time Range to a Numbered ACL

```
Router(config)# time-range smtp
Router(config-time-range)# periodic daily 5:00 to 23:59
Router(config)# access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255
established
Router(config)# access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq
25 time-range smtp
Router(config)# interface serial 0
Router(config-if)# ip access-group 102 in
```

Applying a Time Range to a Named Access Control List

To apply a time range to a named extended access control list (ACL), enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list {standard | extended} access-list-name`	Defines an access list by name and enters named-access-control configuration mode. **Note** The time-based ACLs feature supports only extended access lists.
Step 2	`Router(config-ext-nacl)# {deny | permit} protocol source source-wildcard destination destination-wildcard [icmp-type [icmp-code] | icmp-message] [precedence precedence] [tos tos] [log] time-range time-range-name [fragments]`	Sets conditions in a named IP access list that will deny or permit packets. The **time-range** *time-range-name* option indicates the name of the time range that applies to this ACE. **Note** In Cisco IOS Release 12.3(7)XI1, the **time-range** argument is required.
Step 3	`Router(config)# interface type number slot/module/port.subinterface`	Configures an interface and enters interface configuration mode.
Step 4	`Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}`	Controls access to an interface.

Example 12-5 denies FTP traffic on Monday through Sunday between the hours of 9:00 a.m. and 3:00 p.m. The example creates the time range named *no-ftp* and applies it to the ACE of the extended IP access list named I. The time-based ACL is then applied to the ingress Ethernet 0 interface.

Example 12-5 Applying a Time Range to a Named ACL

```
Router(config)# time-range no-ftp
Router(config-time-range)# periodic daily 9:00 to 15:00
Router(config)# ip access-list extended strict
Router(config-ext-nacl)# deny tcp any any eq 21 time-range no-ftp
Router(config-ext-nacl)# exit
Router(config)# interface ethernet 0
Router(config-if)# ip access-group strict in
```

Monitoring and Maintaining Time-Based ACLs

To monitor and maintain time-based ACLs, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show access-lists** [*access-list-number* | *access-list-name*]	Displays the contents of current access lists or the access list you specify.
Router# **show interface** *type number*	Displays information about the interface you specify and indicates if an access list is configured on the interface.
Router# **show time-range**	Displays the configured time ranges.

Configuration Examples for Time-Based ACLs

The following example permits Telnet connections from the 10.1.1.0 network to the 172.16.1.0 network on Monday, Wednesday, and Friday during the business hours.

```
time-range EVERYOTHERDAY
 periodic Monday Wednesday Friday 8:00 to 17:00
!
access-list 101 permit tcp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 eq telnet time-range
EVERYOTHERDAY
!
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 101 in
```

The following example permits SMTP traffic from all networks to indefinitely access all networks beginning at 12:00 p.m. on January 1, 2001.

```
time-range forever
 absolute start 12:00 1 January 2001
!
ip access-list extended allusers
 permit tcp any any eq 25 time-range forever
```

The following example permits UDP traffic until noon on December 31, 2000. The ACL entry will no longer allow UDP traffic after that date and time.

```
time-range stop-udp
 absolute end 12:00 31 December 2000
!
ip access-list extended usa
 permit udp any any time-range stop-udp
```

The following configuration example permits telnet traffic on Monday, Tuesday, and Friday from 9:00 a.m. and 5:00 p.m.:

```
time-range telnet
  periodic Monday Tuesday Friday 9:00 to 17:00
!
ip access-list extended camden
  permit tcp any any eq telnet time-range telnet
```

The following configuration example permits UDP traffic on Saturday and Sunday from 8:00 a.m. on January 1, 1999 to 6:00 p.m. on December 31, 2001:

```
time-range udp
  absolute start 8:00 1 January 1999 end 18:00 31 December 2001
  periodic weekends 00:00 to 23:59
!
ip access-list extended boothbay
  permit udp any any time-range udp
```

CHAPTER 13

Unicast Reverse Path Forwarding

Cisco integrated security systems incorporate a comprehensive selection of feature-rich security services, offering commercial, enterprise and service provider customers the ability to deploy trusted and protected business applications and services.

Threat defense is a critical aspect of an integrated security approach and involves the implementation of proactive measures. One valuable threat defense tool is unicast Reverse Path Forwarding (uRPF).

The key function of uRPF is to verify that the path of an incoming packet is consistent with the local packet forwarding information. This is achieved by performing a reverse path look-up (hence the feature's name) using the source IP address of an incoming packet to determine the current path (adjacency) to that IP address. The validity of this path determines whether uRPF passes or drops the packet.

The specific uRPF path validation criteria that is used to determine path consistency is dependent upon the particular uRPF mode enabled on an interface. Table 13-1 shows two uRPF modes which are supported by Cisco 10000 series routers.

Table 13-1 Three uRPF Modes

uRPF Mode	Path Resolution Table	uRPF Path Selection Criteria
Strict	CEF FIB	Path to the source IP address must be through the SAME interface as that on which the packet arrived
Loose	CEF FIB	Path to the source IP address is through *any* interface on the device

If the path is:

- Valid—the packet will be passed.
- Invalid—the packet is silently discarded.

uRPF uses the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB) to perform reverse path look-up on the source IP address of an incoming packet. The CEF FIB is a database of network layer routing information and associated forwarding/adjacency information used in the CEF switching of packets. The CEF FIB is populated with the path for all known IP prefixes and their associated adjacencies. It is thus a key element of uRPF reverse path validation. After enabled on an interface, uRPF checks all IP packets on the input path of that interface.

Note Cisco 10000 series routers support both strict and loose mode uRPF for IPv4. However, for IPv6, the router supports only strict uRPF.

The uRPF feature is described in the following topics:

Feature History for uRPF

Cisco IOS Release	Description	Required PRE
12.2(27)SBB	This feature was introduced on the Cisco 10000 series router with strict mode only.	PRE2
12.2(33) SB	This feature was integrated on Cisco 10000 with both strict and loose modes for IPv4 traffic.	PRE2, PRE3, and PRE4

Prerequisites for uRPF

Before you configure uRPF on a router, ensure that the interface supports IP addressing. For a broadband interface, uRPF configurations must be added in the virtual template with all of the other IP configurations.

Restrictions for uRPF

The uRPF feature in Cisco 10000 has the following restrictions:

- Clients should not be multihomed to the same router because multihoming defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.
- Unicast RPF is available only for platform images that support CEF. Unicast RPF is supported in Cisco IOS Releases 11.1(17)CC, 12.0, and later. It is not available in Cisco IOS Release 11.2 or 11.3.
- Unicast RPF is not supported by MPLS. It is supported only by IP traffic—IPv4 and IPv6. However, IPv6 supports uRPF in strict mode only, with the allow-default option on.
- Unicast RPF does not support access control lists (ACLs).
- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the router. For more information about CEF, see the *Cisco IOS Switching Services Configuration Guide*.

- By default, without uRPF provision urpf drops can be seen in pxf when:
  - the interface is not up
  - there is no ip address on the interface

Configuring Unicast RPF

To use Unicast RPF, you must configure the router for CEF switching or CEF distributed switching. There is no need to configure the input interface for CEF switching because Unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation and operates on IP packets received by the router. It is very important that CEF be turned on globally in the router—Unicast RPF will not work without CEF.

To configure Unicast RPF, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **ip cef**	Enables CEF on the router. You might want to disable CEF on a particular interface if that interface is configured with a feature that CEF does not support. You can enable CEF globally, but disable CEF on a specific interface by using the **no ip route-cache cef** interface command that enables all but that specific interface to use express forwarding. If you have disabled CEF operation on an interface and want to reenable it, you can use the **ip route-cache cef** command in interface configuration mode.
Step 2	Router(config-if)# **interface** *type*	Selects the input interface on which you want to apply Unicast RPF. This is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding the packet on to the next destination. The interface type is specific to your router and the types of interface cards installed on the router. To display a list of available interface types, enter the **interface ?** command.
Step 3	Router(config-if)# **ip verify unicast source reachable-via any** or Router(config-if)# **ip verify unicast source reachable-via rx**	Enables Unicast RPF on the interface. The **any** option enables a Loose Mode uRPF on the router. This mode allows the router to reach the source address via any interface. The **rx** option enables a Strict Mode uRPF on the router. This mode ensures that the router reaches the source address only via the interface on which the packet was received. You can also use the **allow-default** option, so that the default route can match when checking source address. The **allow-self-ping option** allows the router to ping itself.
Step 4	Router(config-if)# **exit**	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Note You can use default route to configure a default path for all addresses that are not in the regular routing table. When configuring uRPF, you can use the allow-default option to allow ip packets with the source address resolved to a valid default path, depending on the uRPF modes. In strict mode uRPF, the packets are allowed from the same interface that has been pointed by the default route. In loose mode uRPF, packets with the source address resolved to the default route are allowed. However, if there is no default route provisioned in the router, the allow-default option on or off would not make any difference regardless of the uRPF mode as there is no valid default path.

Monitoring and Maintaining uRPF

Unicast RPF counts the number of packets dropped or suppressed because of malformed or forged source addresses. Unicast RPF counts dropped or forwarded packets that include the following global and per-interface information:

- Global Unicast RPF drops
- Per-interface Unicast RPF drops

After you enable uRPF on a router, you can monitor the number of packets getting dropped by the router using the following commands.

Command	Description
Router# **show ip traffic**	Displays global router statistics about Unicast RPF drops and suppressed drops.
Router# **show ip interface** *type*	Displays per-interface statistics about Unicast RPF drops and suppressed drops.
Router# **show pxf cpu statistics drop** *interface*	Displays drop counters by pxf for a given interface, even without uRPF provision and if the interface is not up or does not have an IP address.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Example 13-1 shows the total number (global count) of dropped packets for all interfaces on the router using the **show ip traffic** command. The Unicast RPF drop count is included in the IP statistics section.

Example 13-1 show ip traffic Command

```
Router# show ip traffic

IP statistics:
  Rcvd:  1753234 total, 1163482 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         1162010 unknown protocol, 523362 not a gateway
         0 security failures, 0 bad options, 0 with options
```

```
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 331512 received, 0 sent
  Mcast: 0 received, 0 sent
  Sent:  15 generated, 0 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 5 unicast RPF, 0 forced drop, 0 unsupported-addr
         0 options denied, 0 source IP address zero
```

A nonzero value for the count of dropped or suppressed packets can mean one of two things:

- Packets have a bad source address (normal operation).
- Router is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

Note The RPF counter increases when the source address resolves to a NULL 0 because the address is then considered as spoof.

Example 13-2 shows the total of dropped or suppressed packets at a specific interface using the **show ip interface** command.

Example 13-2 show ip interface Command

```
Router> show ip interface gigabitEthernet 8/1/0

GigabitEthernet8/1/0 is up, line protocol is up
  Internet address is 80.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF switching turbo vector
  IP CEF turbo switching turbo vector
  Associated unicast routing topologies:
        Topology "base", operation state is UP
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
```

```
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  BGP Policy Mapping is disabled
  Input features: uRPF
  IP verify source reachable-via ANY
   5 verification drops
   5 suppressed verification drops
   0 verification drop-rate
```

Example 13-3 shows how uRPF drops can also be seen at the PXF using the **show pxf cpu statistics drop** *interface* command.

Example 13-3 show pxf cpu statistics drop interface Command

```
router# sh pxf cpu statistics drop g8/1/0
FP drop statistics for GigabitEthernet8/1/0
                            packets            bytes
    vcci undefined          0                  0
    bad vlan id             0                  0
  vcci 9E6
    in l2 max mtu           0                  0
    in l2 min mtu           0                  0
    encap not supported     0                  0
    mlfr fragament          0                  0
    mpls not enabled        0                  0
    ip version              0                  0
    ip header length        0                  0
    ip length max           0                  0
    ip length min           0                  0
    ip checksum             0                  0
    fib rpf fail            0                  0
    acl denied              0                  0
    ttl                     0                  0
    unreachable             0                  0
    df multicast            0                  0
    police input drop       0                  0
    police output drop      0                  0
    out l2 max mtu          0                  0
    out l2 min mtu          0                  0
    tunnel no match         0                  0
    iedge input drop(s)     0                  0
    iedge output drop(s)    0                  0
```

Configuration Examples of uRPF

This section provides the following configuration examples:

- Configuring Loose Mode uRPF
- Configuring Loose Mode uRPF with the allow-self-ping Option
- Configuring Loose Mode uRPF with the allow-default Option

Configuring Loose Mode uRPF

Example 13-4 shows how to enable Loose Mode uRPF on a router over the Gigabit Ethernet Interface:

Example 13-4 Loose Mode uRPF configuration on 8/1/0 interface

```
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router (config)# int g8/1/0
Router (config-if)# ip verify unicast source reachable-via?
  any Source is reachable via any interface
  rx   Source is reachable via interface on which packet was received

Router (config-if)# ip verify unicast source reachable-via any?
  <1-199>          IP access list (standard or extended)
  <1300-2699>      IP expanded access list (standard or extended)
  allow-default    Allow default route to match when checking source address
  allow-self-ping  Allow router to ping itself (opens vulnerability in
                   verification)
  <cr>

Router (config-if)# ip verify unicast source reachable-via any
Router (config-if)# end
```

Example 13-5 shows how you can use the **show router interface** command for verifying that Loose Mode uRPF has been configured on a router

Example 13-5 Verifying Loose Mode uRPF on 8/1/0 interface

```
Router# sh ru interface gig8/1/0
!
interface GigabitEthernet8/1/0
 ip address 80.1.1.1 255.255.255.0
 ip verify unicast source reachable-via any
 negotiation auto
end
```

Configuring Loose Mode uRPF with the allow-self-ping Option

Example 13-6 shows how you can configure Loose Mode uRPF with the **allow-self-ping** option.

Example 13-6 Loose Mode uRPF with the allow-self-ping option

```
Router(config)# int g8/1/0
Router(config-if)# ip verify unicast source reachable-via any allow-self-ping
Router(config-if)# end
Router# sh ru int g8/1/0
!
interface GigabitEthernet8/1/0
 ip address 80.1.1.1 255.255.255.0
 ip verify unicast source reachable-via any allow-self-ping
 negotiation auto
end
```

Note After you enable the interface with uRPF using the allow-self ping option, initiate a self-ping to see whether the self-ping option is successful.

Configuring Loose Mode uRPF with the allow-default Option

Example 13-7 shows how you can configure Loose Mode uRPF with the **allow-default** option.

Example 13-7 Loose Mode uRPF with the allow-default option

```
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int g8/1/0
Router(config-if)# ip verify unicast source reachable-via any allow-default
Router(config-if)# end
Router# sh ru int gig8/1/0
!
interface GigabitEthernet8/1/0
 ip address 80.1.1.1 255.255.255.0
 ip verify unicast source reachable-via any allow-default
 negotiation auto
end
```

Note For configuring Strict mode uRPF, replace the **any** keyword with **rx** in the **ip verify unicast source reachable-via** command.

CHAPTER 14

Configuring Automatic Protection Switching

Automatic protection switching (APS) is a protection mechanism for SONET networks that enables SONET connections to switch to another SONET circuit when a circuit failure occurs. A protect interface serves as the backup interface for the working interface. When the working interface fails, the protect interface quickly assumes its traffic load. This chapter describes the following APS features:

Multirouter Automatic Protection Switching

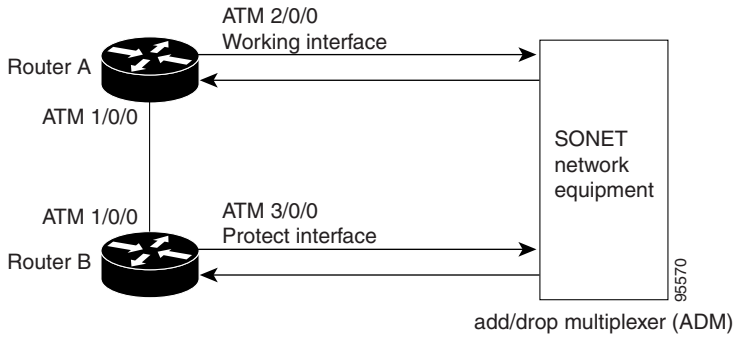
The Multirouter Automatic Protection Switching (MR-APS) feature enables interface connections to switch from one circuit to another circuit if a circuit failure occurs. Interfaces can be switched in response to a router failure, degradation or loss of channel signal, or manual intervention. In a multirouter environment, the Multirouter APS (MR-APS) feature allows the protect SONET interface to reside in a different router from the working SONET interface.

The protection mechanism used for this feature has a linear 1+1 architecture as described in the Bellcore publication TR-TSY-000253, SONET Transport Systems; Common Generic Criteria, Section 5.3. The connection may be bidirectional, and revertive or nonrevertive. Unidirectional MR-APS is not supported. The default is bidirectional. The switching mode must be the same on the far end of the connection.

In the 1+1 architecture, a protect interface (circuit) is paired with each working interface. Normally, the protect and working interfaces are connected to an ADM (add/drop multiplexer), which sends the same signal payload to the working and protect interfaces.

Figure 1 shows a multirouter APS configuration. In the figure, the working and protect circuits terminate on different line cards that are installed in two different routers. Interfaces in a multirouter APS configuration can be configured with either SONET or SDH framing.

Figure 14-1 Multirouter APS Configuration



On the protect circuit, the K1 and K2 bytes from the line overhead (LOH) of the SONET frame indicate the current status of the APS connection and convey any requests for action. This signalling channel is used by the two ends of the connection to maintain synchronization.

The working and protect circuits themselves, within the router or routers in which they terminate, are synchronized over an independent communication channel, not involving the working and protect circuits. In Figure 14-1, this independent channel may be a different ATM connection or a lower-bandwidth connection. In a router configured for multirouter APS, the configuration for the protect interface includes the IP address of the router (normally its loopback address) that has the working interface.

This chapter describes the MR-APS feature in the following topics:

Feature History for MR-APS

Cisco IOS Release	Description	Required PRE
12.0(23)SX	This feature was introduced on the Cisco 10000 series router.	PRE1
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S.	PRE1
12.3(7)XI2	This feature was integrated into Cisco IOS Release 12.3(7)XI2.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for MR-APS

In Cisco IOS Releases 12.3(7)XI2 and 12.2(28)SB, MR-APS is supported for the following line cards:

- 4-Port OC3/STM-1 ATM line card
- 1-Port OC-12 ATM line card
- 1-Port Channelized OC-12/STM-4 line card
- 4-Port Channelized OC-3/STM-1 line card

In Cisco IOS Release 12.0(26)S, MR-APS is also supported for the following line cards:

- 6-Port OC-3/STM-1 Packet over SONET line card
- 1-Port OC-12 Packet over SONET line card

Configuration Tasks for MR-APS

To configure the MR-APS feature, perform the following tasks:

- Configuring MR-APS on Unchannelized Line Cards, page 14-3
- Configuring MR-APS on Channelized Line Cards, page 14-4
- Configuring MR-APS with Static Routes, page 14-5

Configuring MR-APS on Unchannelized Line Cards

To configure MR-APS on unchannelized line cards, enter the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode, which allows you to associate two line cards as a redundant pair.
Step 2	`Router(config-r)# associate slot` *slot-one* **mr-aps**	Logically associates slots for APS processor redundancy. To allow MR-APS to operate, you must associate a slot on the working interface of one router and with a corresponding protect interface on a second router.
Step 3	`Router(config-r)# exit`	Exits redundancy configuration mode and returns to global configuration mode.
Step 4	`Router(config)# interface` *type number*	Specifies the interface type and number. Enters interface configuration mode.
Step 5	`Router(config-if)# aps group` *group-number*	Permits more than one APS protect and working interface to be supported on a router.
Step 6	`Router(config-if)# aps working` *circuit-number*	Configures an interface as a working interface.
Step 7	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	Repeat steps 1 through 5 on the second router to configure the protect interface. Substitute the appropriate slot numbers, interface types, and interface numbers. After you complete step 5, go to step 9.	

	Command	Purpose
Step 9	`Router(config-if)# aps protect circuit-number ip-address`	Configures an interface as a protect interface. The *ip-address* argument specifies the IP address of the router that has the working interface.
Step 10	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring MR-APS on Channelized Line Cards

To configure MR-APS on channelized line cards, enter the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode, which allows you to associate two line cards as a redundant pair.
Step 2	`Router(config-r)# associate slot slot-one mr-aps`	Logically associates slots for APS processor redundancy. To allow MR-APS to operate, you must associate a slot on the working interface of one router and with a corresponding protect interface on a second router.
Step 3	`Router(config-r)# exit`	Exits redundancy configuration mode and returns to global configuration mode.
Step 4	`Router(config)# controller SONET slot#/subslot#/port#`	Specifies the interface type and number. Enters controller configuration mode.
Step 5	`Router(config-controller)# aps group group-number`	Permits more than one APS protect and working interface to be supported on a router.
Step 6	`Router(config-controller)# aps working circuit-number`	Configures an interface as a working interface.
Step 7	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.
Step 8	Repeat steps 1 through 5 on the second router to configure the protect interface. Substitute the appropriate slot numbers, interface types, and interface numbers. After you complete step 5, go to step 9.	
Step 9	`Router(config-controller)# aps protect circuit-number ip-address`	Configures an interface as a protect interface. The *ip-address* argument specifies the IP address of the router that has the working interface.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.

Example 14-1 shows the configuration of MR-APS on ATM interfaces. In the example, Router A is configured with the working interface, and Router B is configured with the protect interface. If the working interface on Router A becomes unavailable, the connection automatically switches over to the protect interface on Router B.

Example 14-1 Configuring MR-APS

Router A (working interface)

```
configure terminal
interface atm 1/0/0
ip address 10.7.7.7 255.255.255.0
!
redundancy
associate slot 2 mr-aps
!
interface atm 2/0/0
aps group 1
aps working 1
```

Router B (protect interface)

```
interface atm 1/0/0
ip address 10.7.7.6 255.255.255.0
!
redundancy
associate slot 3 mr-aps
!
interface atm 3/0/0
aps group 1
aps protect 1 10.7.7.7
```

Configuring MR-APS with Static Routes

To configure MR-APS with static routes, perform the following procedures:

- Configuring MR-APS with Static Routes on Unchannelized Line Cards, page 14-5
- Configuring MR-APS with Static Routes on Channelized Line Cards, page 14-7

Configuring MR-APS with Static Routes on Unchannelized Line Cards

To optionally configure MR-APS with static routes on unchannelized line cards, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode, which allows you to associate two line cards as a redundant pair.
Step 2	`Router(config-r)# associate slot` *slot-one* **mr-aps**	Logically associates slots for APS processor redundancy. To allow MR-APS to operate, you must associate a slot on the working interface of one router and with a corresponding protect interface on a second router.
Step 3	`Router(config-r)# exit`	Exits redundancy configuration mode and returns to global configuration mode.

	Command	Purpose
Step 4	`Router(config)# ip route prefix mask {ip-address | interface-type interface-number [ip-address]} [distance] [name] [permanent] [tag tag]`	Configures a static IP address. When configuring APS, we recommend that you specify the optional IP address of the interface to improve routing performance.
Step 5	`Router(config)# interface type number`	Specifies the interface type and number. Enters interface configuration mode or controller configuration mode.
Step 6	`Router(config-if)# ip route static update immediate`	(Optional) Specifies that static routes will be added to the routing table immediately after the interface becomes active.
Step 7	`Router(config-if)# carrier-delay [seconds | msec seconds]`	Sets the carrier delay timer value in seconds or milliseconds. This command allows you to filter link outages and to not report them as a link down event if they occur before the carrier delay timer expires. In MR-APS, system performance can be enhanced if link-down event messages are kept to a minimum.
Step 8	`Router(config-if)# aps group group-number`	Permits more than one APS protect and working interface to be supported on a router.
Step 9	`Router(config-if)# aps working circuit-number`	Configures an interface as a working interface.
Step 10	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	Repeat steps 1 through 8 on the second router to configure the protect interface. Substitute the appropriate slot numbers, IP addresses, interface types, and interface numbers. After you complete step 8, go to step 12.	
Step 12	`Router(config-if)# aps protect circuit-number ip-address`	Configures an interface as a protect interface. The *ip-address* argument specifies the IP address of the router that has the working interface.
Step 13	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring MR-APS with Static Routes on Channelized Line Cards

To optionally configure MR-APS with static routes on channelized line cards, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode, which allows you to associate two line cards as a redundant pair.
Step 2	`Router(config-r)# associate slot slot-one mr-aps`	Logically associates slots for APS processor redundancy. To allow MR-APS to operate, you must associate a slot on the working interface of one router and with a corresponding protect interface on a second router.
Step 3	`Router(config-r)# exit`	Exits redundancy configuration mode and returns to global configuration mode.
Step 4	`Router(config)# ip route prefix mask {ip-address | interface-type interface-number [ip-address]} [distance] [name] [permanent] [tag tag]`	Configures a static IP address. When configuring APS, we recommend that you specify the optional IP address of the interface to improve routing performance.
Step 5	`Router(config)# controller SONET slot#/subslot#/port#`	Specifies the interface type and number. Enters controller configuration mode.
Step 6	`Router(config-controller)# ip route static update immediate`	(Optional) Specifies that static routes will be added to the routing table immediately after the interface becomes active.
Step 7	`Router(config-controller)# carrier-delay [seconds | msec seconds]`	Sets the carrier delay timer value in seconds or milliseconds. This command allows you to filter link outages and to not report them as a link down event if they occur before the carrier delay timer expires. In MR-APS, system performance can be enhanced if link-down event messages are kept to a minimum.
Step 8	`Router(config-controller)# aps group group-number`	Permits more than one APS protect and working interface to be supported on a router.
Step 9	`Router(config-controller)# aps working circuit-number`	Configures an interface as a working interface.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.
Step 11	Repeat steps 1 through 8 on the second router to configure the protect interface. Substitute the appropriate slot numbers, IP addresses, interface types, and interface numbers. After you complete step 8, go to step 12.	
Step 12	`Router(config-controller)# aps protect circuit-number ip-address`	Configures an interface as a protect interface. The *ip-address* argument specifies the IP address of the router that has the working interface.
Step 13	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.

Example 14-2 shows the configuration of multirouter APS with static routes on ATM interfaces. Router A is configured with the working interface, and Router B is configured with the protect interface. If the working interface on Router A becomes unavailable, the connection automatically switches over to the protect interface on Router B. Note that 172.16.1.0 is the address of the traffic destination network and that the route over the Peer Group Protocol (PGP) link has a higher distance metric number than the multirouter APS working interface.

Example 14-2 Configuring MR-APS with Static Routes

Router A (working interface)

```
configure terminal
interface atm 1/0/0
ip address 10.7.7.7 255.255.255.0
ip route static update immediate
carrier-delay msec 8
!
redundancy
associate slot 2 mr-aps
!
interface atm 2/0/0
aps group 1
aps working 1
ip route static update immediate
carrier-delay msec 8
!
ip route 172.16.1.0 255.255.255.0 atm 2/0/0 10
ip route 172.16.1.0 255.255.255.0 atm 1/0/0 10.7.7.6 20
```

Router B (protect interface)

```
configure terminal
interface atm 1/0/0
ip address 10.7.7.6 255.255.255.0
ip route static update immediate
carrier-delay msec 8
!
redundancy
associate slot 3 mr-aps
!
interface atm 3/0/0
aps group 1
aps protect 1 10.7.7.7
ip route static update immediate
carrier-delay msec 8
!
ip route 172.16.1.0 255.255.255.0 atm 3/0/0 10
ip route 172.16.1.0 255.255.255.0 atm 1/0/0 10.7.7.7 20
```

Monitoring and Maintaining the MR-APS Configuration

To monitor and maintain the configuration of MR-APS, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show aps`	Displays about APS-configured interfaces.
`Router# debug aps`	Displays debugging information related automatic protection switching.
`Router(config-if)# aps force` *circuit-number* (unchannelized line cards) or `Router(config-controller)# aps force` *circuit-number* (channelized line cards)	Manually switches the specified circuit to a protect interface, unless a request of equal or higher priority is in effect. *circuit-number* is the number of the circuit to switch to the protect interface. **Note** This command has no effect if the protection channel is already the active channel.
`Router(config-if)# aps manual` *circuit-number* (unchannelized line cards) or `Router(config-controller)# aps manual` *circuit-number* (channelized line cards)	Manually switches a circuit to a protect interface.
`Router# aps lockout [POS | SONET]` *slot#/subslot#/port#*	Prevents a channel from automatically switching to the active, working, or protection state.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Single-router Automatic Protection Switching

The Cisco 10000 series router supports SONET Single-router Automatic Protection Switching (SR-APS) redundancy for the OC-3 ATM, OC-12 ATM, OC-12 POS, 6-port OC-3 POS, channelized OC-12, and channelized 4-port STM-1 line cards. The following types of SR-APS are supported:

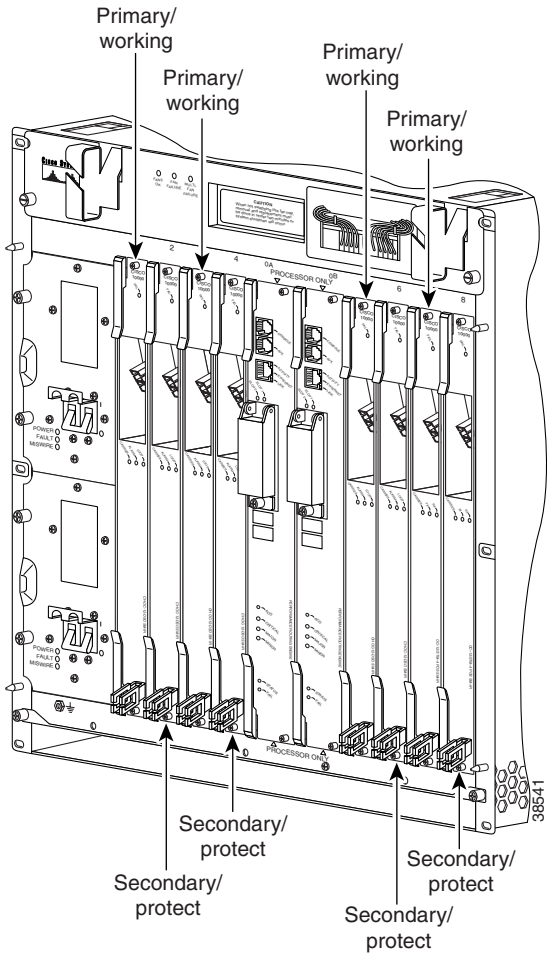
- SR-APS 1+1 support for line cards with a single port (such as the OC-12 POS) is card-to-card. When the active line card fails, the redundant line card takes over.
- SR-APS 1:1 support for line cards with multiple ports (such as the OC-3 POS) is port-to-port. The PRE transmits data to both the active and the redundant line card. When a port fails on the active line card, the corresponding port on the redundant line card takes over. In addition to port failovers, multiple port line cards support line card failover. If the working card fails, the protect card becomes active and all ports on that card are active.

When you associate slots, the software pairs an odd-numbered slot with the next higher even-numbered slot:

- Odd-numbered slot—Holds the primary card, or working card
- Even-numbered slot—Holds the secondary card, or protect card

Figure 14-2 shows the redundant slot pairings in the Cisco 10008 chassis.

Figure 14-2 Redundant Slot Pairings in the Cisco 10008 Series Router



Note Only slots 1 and 2 and slots 3 and 4 in the Cisco 10005 chassis can be used for APS redundancy because slot 5 does not have an associated higher, even-numbered slot.

This chapter describes the SR-APS feature in the following topics:

Feature History for SR-APS

Cisco IOS Release	Description	Required PRE
12.0(21)ST	This feature was introduced on the Cisco 10000 series router.	PRE1
12.2(13)BZ	This feature was integrated into Cisco IOS Release 12.2(13)BZ	PRE1
12.3(7)XI	This feature was integrated into Cisco IOS Release 12.3(7)XI.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Configuring SR-APS

To configure SR-APS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode, which allows you to associate two line cards as a redundant pair.
Step 2	`Router(config-r)# associate slot` odd-*slot* *even-slot*	Associates two line cards as a redundant pair. To allow SR-APS to operate, you must specify a line card installed in an odd slot number as the first member of a redundant pair; the second line card must be installed in the even-numbered slot to its right.
Step 3	`Router(config-r)# exit`	Exits redundancy configuration mode and returns to global configuration mode.

Note After you configure redundancy, the software treats the pair as though it occupies a single slot. The interface slot number is always the odd number of the redundant pair. For example, for the redundant pair occupying slots 5 and 6, the **show interface pos 5/0/0** command refers to the active card (even if the active card occupies slot 6).

Disabling SR-APS

To disable SR-APS redundant operation, use the **no** form of the **associate slot** command. For example:

```
Router(config-r)# no associate slot 3 4
```

If the redundant configuration is disabled, the software modifies the running configuration in the following ways:

1. The software removes all SR-APS configuration information.
2. The software creates two configurations, one for the primary card and one for the protect card.

Table 14-1 shows examples of configuration files with redundancy enabled and disabled.

Table 14-1 Configuration File—Redundancy Enabled and Disabled

Redundancy Enabled	Redundancy Disabled
card 5/0 1oc12pos-1 card 6/0 1oc12pos-1 ! redundancy associate slot 5 6 ! interface POS5/0/0 ip address 5.5.5.5 255.255.255.0 no ip directed-broadcast ip mtu 1500 loopback internal no keepalive aps mode linear 1+1 nonreverting unidirectional aps signal-fail BER threshold 3 aps signal-degrade BER threshold 5 crc 32 clock source internal pos scramble-atm pos threshold sd-ber 5 pos flag c2 0 pos flag j0 0	card 5/0 1oc12pos-1 card 6/0 1oc12pos-1 ! interface POS5/0/0 ip address 5.5.5.5 255.255.255.0 no ip directed-broadcast ip mtu 1500 loopback internal no keepalive crc 32 clock source internal pos scramble-atm pos threshold sd-ber 5 pos flag c2 0 pos flag j0 0 ! interface POS6/0/0 ip address 6.6.6.6 255.255.255.0 no ip directed-broadcast ip mtu 1500 no ip route-cache cef no keepalive …

Monitoring and Maintaining the SR-APS Configuration

To monitor and maintain the configuration of SR-APS, enter any of the following commands in privileged EXEC mode:

Command	Purpose
Router# **show aps**	Displays the status of the APS-configured slot.
Router# **aps clear** [**POS** | **SONET**] *slot#/subslot#/port#*	Clears the SR-APS commands on a channel and enables automatic SR-APS switching.
Router# **aps force** [**POS** | **SONET**] *slot#/subslot#/port#* **from** [**working** | **protection**]	Forces a switch from the working channel to the protection channel, or from the protection channel to the working channel. **from working**—Forces a switch from the working channel to the protection channel. **Note** This command has no effect if the protection channel is currently the active channel. **from protection**—Forces a switch from the protection channel to the working channel. **Note** This command has no effect if the working channel is currently the active channel.

Command	Purpose
`Router# aps lockout [POS | SONET] slot#/subslot#/port#`	Prevents a channel from automatically switching to the active, working, or protection state.
`Router# aps manual [POS | SONET] slot#/subslot#/port# from [working | protection]`	Manually switches from the working channel to the protection channel, or from the protection channel to the working channel. **from working**—Manually switches from the working channel to the protection channel. **Note** This command has no effect if the protection channel is currently the active channel. **from protection**—Manually switches from the protection channel to the working channel. **Note** This command has no effect if the working channel is currently the active channel.

Example 14-3 Clearing SR-APS Commands on a Channel

Example 14-3 shows how to clear SR-APS commands on redundant Channelized OC-12 POS cards in slots 5 and 6:

```
Router# aps clear pos 5/0/0
```

Example 14-4 Forcing a SR-APS Switch

Example 14-4 shows how to force a switch from the working channel to the protection channel:

```
Router# aps force POS 5/0/0 from working
```

Example 14-5 Performing a Manual SR-APS Switch

Example 14-5 show how to manually switch the active channel from the working channel to the protection channel:

```
Router# aps manual POS 5/0/0 from working
```

Threshold Commands

Threshold commands allow you to specify criteria that trigger a cutover. In addition to the criteria set by these commands, cutovers are triggered by Section Loss of Signal (SLOS) critical alarms, Section Loss of Frame (SLOF) critical alarms, and Line Alarm Indicate Signal (LAIS) major alarms.

Specifying SR-APS Signal Degrade BER Threshold

Use the **aps signal-degrade BER threshold** command to modify the bit error rate threshold that, if exceeded, triggers an APS cutover.

```
aps signal-degrade BER threshold value
[no] aps signal-degrade
```

Where *value* can be in the range of 10^{-5} to 10^{-9}. Enter this value as a single digit between 5 and 9.

The default signal degrade BER threshold value is 10^{-6}.

Use the **no** form of the command to return the threshold value to its default.

In the following example, the threshold value is set to 10^{-8}.

```
Router(config)# interface pos 8/0/0
Router(config-if)# aps signal-degrade BER threshold 8
```

Specifying SR-APS Signal Fail BER Threshold

Use the **aps signal-fail BER threshold** command to modify the bit error rate threshold that, if exceeded, causes an APS cutover.

aps signal-fail BER threshold *value*
[**no**] **aps signal-degrade**

Where *value* can be in the range of 10^{-3} to 10^{-5}. Enter this value as a single digit between 3 and 5.

The default signal fail BER threshold value is 10^{-3}.

Use the **no** form of the command to return the threshold value to its default.

In the following example, the threshold value is set to 10^{-4}:

```
Router(config)# interface pos 8/0/0
Router(config-if)# aps signal-fail BER threshold 4
```

CHAPTER 15

Configuring IP Multicast

The IP multicast feature enables a host to send packets to a subset of hosts known as a multicast group. The hosts in the multicast group are the group members. Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability. Any host, regardless of whether it is a member of a group, can send messages to a group. However, only the members of a group can receive the messages.

Enhancements to the IP multicast feature provide support for broadband environments. This enhanced IP multicast feature allows PPPoA, PPPoE, and RBE subscribers to participate in multicast groups and to initiate multicast messages.

The IP multicast feature supports the following protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP)—Used between hosts on a LAN and the router(s) on that LAN to track the multicast groups of which hosts are members.
- Protocol-Independent Multicast (PIM)—Used between routers so that they can track which multicast packets to forward to each other to their directly connected LANs.
- Distance Vector Multicast Routing Protocol (DVMRP)—Used on the multicast backbone of the Internet. The Cisco IOS software supports PIM-to-DVMRP interaction. However, you cannot run DVMRP back-to-back between Cisco routers.
- Cisco Group Management Protocol (CGMP)—Used on routers connected to Cisco Catalyst switches to perform tasks similar to those performed by IGMP.

Note For more information about the IP multicast feature, see the "IP Multicast" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

This chapter describes the IP Multicast feature in the following topics:

Feature History for IP Multicast

Cisco IOS Release	Description	Required PRE
12.2(4)BZ1	This feature was introduced on the Cisco 10000 series router.	PRE1
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7) XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for IP Multicast

The IP Multicast feature has the following restrictions:

- The Cisco 10000 series router software does not support running Distance Vector Multicast Routing Protocol (DVMRP) back to back between routers.
- If you enable IP multicast fast switching on one interface, you must enable it on all outbound interfaces on the router. Failure to do so results in the router sending duplicate multicast packets out the interface that has fast switching enabled.

Configuration Tasks for IP Multicast Routing

To configure basic IP multicast routing, perform the following tasks:

- Enabling IP Multicast Routing, page 15-2
- Enabling PIM on an Interface, page 15-3
- Enabling Dense Mode, page 15-3
- Enabling Sparse Mode, page 15-3
- Enabling Sparse-Dense Mode, page 15-4
- Configuring Native Multicast Load Splitting, page 15-4

For information on other optional basic and advanced tasks, see the “IP Multicast” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Enabling IP Multicast Routing

IP multicast routing allows the Cisco IOS software to forward multicast packets. To enable IP multicast routing on the Cisco 10000 router, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# ip multicast-routing`	Enables IP multicast routing.

Note If you enable IP multicast fast switching on one interface, you must enable it on **all** outbound interfaces on the router. Failure to do so results in the router sending duplicate multicast packets out the interface that has fast switching enabled.

Enabling PIM on an Interface

The protocol-independent multicast (PIM) protocol maintains the current IP multicast service mode of receiver initiated membership. Enabling PIM on an interface also enables IGMP operation on that interface. Configure an interface in one of the following modes:

- Dense mode
- Sparse mode
- Sparse-dense mode

The mode determines how the Cisco 10000 router populates its multicast routing table and how it forwards multicast packets it receives from its directly connected LANs. You must enable PIM in one of these modes for an interface to perform IP multicast routing.

For more information, see the "IP Multicast" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Enabling Dense Mode

To configure PIM on an interface to be in dense mode, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip pim dense-mode`	Enables dense mode PIM on the interface.

Enabling Sparse Mode

To configure PIM on an interface to be in sparse mode, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip pim sparse-mode`	Enables sparse mode PIM on the interface.

Enabling Sparse-Dense Mode

When you enable sparse-dense mode, the interface is treated as dense mode if the group is in dense mode. If the group is in sparse mode, the interface is treated in sparse mode. You must have a rendezvous point (RP) if the interface is in sparse-dense mode and you want to treat the group as a sparse group. For more information, see the "IP Multicast" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

To enable PIM to operate in the same mode as the group, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip pim sparse-dense-mode`	Enables PIM to operate in sparse or dense mode, depending on the multicast group.

Configuring Native Multicast Load Splitting

You can configure multicast traffic from different sources to be load split across equal cost paths to take advantage of multiple paths through the network.

For more information about configuring native multicast load splitting, see the configuration document, located at the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a00805a595a.html

Note A caveat exists for Cisco 10000 series routers; you should not configure native multicast load splitting for PE devices running EIBGP, as this can result in a loss of traffic.

CHAPTER 16

Configuring RADIUS Features

This chapter describes the following features:

RADIUS Attribute Screening

The RADIUS Attribute Screening feature allows you to configure a list of "accept" or "reject" RADIUS attributes on the Cisco 10000 router for authorization and accounting purposes. Based on the accept or reject list you configure for a particular purpose, the Cisco 10000 series router:

- Accepts and processes all standard RADIUS attributes
- Rejects all standard RADIUS attributes

Before you configure a RADIUS accept or reject list, enable AAA using the **aaa new-model** command in global configuration mode. For more information, see the *Cisco IOS Command Summary, Volume 2 of 3, Release 12.2*.

The Cisco 10000 series router supports the RADIUS Attribute Screening feature in the following deployment models:

- Managed L2TP Network Server
- PPP Terminated Aggregation (PTA) to VRF
- Remote Access (RA) to MPLS VPN

Note For more information about RADIUS attribute screening, see the *RADIUS Attribute Screening* feature module.

The RADIUS Attribute Screening feature is described in the following topics:

Feature History for RADIUS Attribute Screening

Cisco IOS Release	Description	Required PRE
12.2(16)BX3	This feature was introduced on the Cisco 10000 series router.	PRE2
12.3(7)XI6	This feature was integrated into Cisco IOS Release 12.3(7) XI6.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for RADIUS Attribute Screening

The following restrictions apply to the RADIUS Attribute Screening feature:

- Network Access Server (NAS) Requirement

  To enable the RADIUS Attribute Screening feature, you should configure the Cisco 10000 router, acting as the NAS, for authorization with RADIUS groups.

- Accept or Reject Lists Limitations

  The two filters used to configure accept or reject lists are mutually exclusive; therefore, you can configure only one accept list or one reject list for each purpose and for each server group.

- Vendor-Specific Attributes

  The RADIUS Attribute Screening feature does not support vendor-specific attribute (VSA) screening. However, you can specify attribute 26 (Vendor-Specific) in an accept or reject list, which will accept or reject all VSAs.

- Required Attributes

  Required attributes in a reject list are allowed to pass through. Do not reject the following required attributes:

  - Authorization—6 (Service-Type) and 7 (Framed-Protocol)
  - Accounting—4 (NAS-IP-Address), 40 (Acct-Status-Type), 41 (Acct-Delay-Time), and44 (Acct-Session-ID)

Note When you configure a reject list with required attributes, an error message does not appear because the list does not specify a purpose (authorization or accounting). The server determines if an attribute is required when the attribute's purpose is known.

Prerequisites for RADIUS Attribute Screening

Before you configure a RADIUS accept or reject list, enable AAA using the **aaa new-model** command in global configuration mode. For more information, see the *Cisco IOS Command Summary, Volume 2 of 3, Release 12.2*.

Configuration Tasks for RADIUS Attribute Screening

To configure and verify the RADIUS Attribute Screening feature, see the "Configuring RADIUS Attribute Accept or Reject Lists" section on page 5-37.

Configuration Examples for RADIUS Attribute Screening

This section provides the following configuration examples:

Authorization Accept Configuration Example

The following example shows how to configure an accept list for attribute 6 (Service-Type) and attribute 7(Framed-Protocol). All other attributes (including VSAs) are rejected for RADIUS authorization.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 10.1.1.1
    authorization accept min-author
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list min-author
    attribute 6-7
```

Accounting Reject Configuration Example

The following example shows how to configure a reject list for attribute 66 (Tunnel-Client-Endpoint) and attribute 67 (Tunnel-Server-Endpoint). All other attributes (including VSAs) are accepted for RADIUS accounting.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 10.1.1.1
    accounting reject tnl-x-endpoint
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list tnl-x-endpoint
    attribute 66-67
```

Authorization Reject and Accounting Accept Configuration Example

The following example shows how to configure a reject list for RADIUS authorization and configure an accept list for RADIUS accounting. Although you cannot configure more than one accept or reject list per server group for authorization or accounting, you can configure one list for authorization and one list for accounting per server group.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 10.1.1.1
    authorization reject bad-author
    accounting accept usage-only
!
radius-server host 10.1.1.1 key mykey1
radius-server attribute list usage-only
    attribute 1,40,42-43,46
!
radius-server attribute list bad-author
    attribute 22,27-28,56-59
```

Rejecting Required Attributes Configuration Example

The following example shows debug output for the **debug aaa accounting** command. In this example, required attributes 44, 40, and 41 have been added to the reject list:

```
Router# debug aaa authorization

AAA/ACCT(6): Accounting method=radius-sg (radius)
RADIUS: attribute 44 cannot be rejected
RADIUS: attribute 61 rejected
RADIUS: attribute 31 rejected
RADIUS: attribute 40 cannot be rejected
RADIUS: attribute 41 cannot be rejected
```

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

RADIUS Transmit Retries

The Cisco 10000 router supports an extended RADIUS transmit retries range. Extending the range of RADIUS transmit retries can protect against lost records if the RADIUS server goes down or communication to it is lost.

You use the **radius-server** command to specify the number of times you want the router to retry transmitting to the RADIUS server. The extended range of values is from 1 to a value higher than 17280.

The RADIUS Transmit Retries feature is described in the following topics:

- Feature History for RADIUS Transmit Retries, page 16-5

Feature History for RADIUS Transmit Retries

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was integrated into Cisco IOS Release 12.3(7) XI1.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Restrictions for RADIUS Transmit Retries

The extended range of RADIUS transmit retries has the following restrictions:

- Using a value at the upper limits of the range of RADIUS transmit retries can force the router to retry for up to 24 hours.
- Using an extended value for RADIUS transmit retries can exhaust the amount of available and allocated buffers.

Configuring RADIUS Transmit Retries

To configure RADIUS transmit retries, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server host {hostname | ip-address} retransmit retries`	Specifies the number of times the router retransmits to the RADIUS server. The *retries* option is a value from 1 to a number greater than 17280.

Note For more information about available options for the **radius-server** command, see the Cisco IOS Command Reference documentation for Cisco IOS Release 12.2.

Configuration Example for RADIUS Transmit Retries

Example 16-1 configures the router to retransmit up to 5 times to the RADIUS server.

Example 16-1 Configuring RADIUS Transmit Retries

```
Router(config)# radius-server host 10.16.1.2 retransmit 5
```

Monitoring and Troubleshooting RADIUS Transmit Retries

To monitor and troubleshoot RADIUS transmit retries, enter any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets. The Number of RADIUS Timeouts field indicates the number of times a server did not respond and the RADIUS server resent the packet.
`Router# debug radius`	Displays detailed information associated with RADIUS.
`Router# debug radius brief`	Displays abbreviated client/server interaction information and abbreviated minimum packet information.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Extended NAS-Port-Type and NAS-Port Support

In Cisco IOS Release 12.3(7)XI1, support for NAS-Port-Type (RADIUS attribute 61), NAS-Port (RADIUS attribute 5), and NAS-Port-ID (RADIUS attribute 87) were changed in the The Extended NAS-Port-Type Attribute Support feature.

The Extended NAS-Port-Type Attribute Support feature is described in the following topics:

Feature History for Extended NAS-Port-Type and NAS-Port Support

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

NAS-Port-Type (RADIUS Attribute 61)

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific Authentication, Authorization, and Accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon. Currently the Internet Engineering Task Force (IETF) RADIUS attributes that are supported include an attribute 61, NAS-Port-Type. NAS-Port-Type indicates the type of physical port the network access server (NAS) is using to authenticate the user.

However there was no method to identify NAS-Port-Type based on a specific broadband service type because the RADIUS RFC does not support extended types that defines these types of ports. Basically all PPPoA, PPPoEoE, and PPPoEoA sessions were identified as being VIRTUAL and all PPPoEoVLAN and PPPoEoQinQ as ETHERNET.

The Extended NAS-Port-Type Attribute Support feature expands NAS-Port-Type, attribute 61, in order that the client can better identify what type of service is taking place on the different types of ports.

One advantage of this feature is that service providers can have their own coding mechanism to track users on given ports differently. Service providers may especially want to track customers using shared resources such as Ethernet or ATM interfaces that have VLANs (or Q-in-Q) and VCs connected to certain customers.

The configuration command **radius-server attribute 61 extended** enables identifying the following new non-RFC compliant, broadband service port types that are indicated by the following numeric values:

- Value 30: PPPoA
- Value 31: PPPoEoA
- Value 32: PPPoEoE
- Value 33: PPPoEoVLAN
- Value 34: PPPoEoQinQ

An additional capability is that subinterfaces such as VLAN, Q-in-Q, VC, or VC ranges are allowed to override the NAS-Port-Type attribute value to be sent on any session that resides on it. This capability provides an extra level of granularity for service providers in managing their end users and allows for further differentiation of different customer usage. This capability is provided with the **radius attribute nas-port-type** [*value*] command.

The value for NAS-Port-Type can be any number chosen by the customer. In particular, customizing your own value is useful when you need to differentiate the NAS-Port-Type based on which type of end client is actually using the port. For example if you want to track mobile clients behind a specific PVC, you can define your own NAS-Port-Type for mobile clients.

NAS-Port (RADIUS Attribute 5)

The NAS-Port (RADIUS attribute 5) is a 32 bit value that uniquely represents the physical or logical port the user is attempting to authenticate on. A logical port can be represented by the virtual path identifier (VPI) and virtual channel identifier (VCI) for an ATM interface, or by the VLAN ID or Q-in-Q ID for an Ethernet interface.

Because each platform and service may have different port information which are relevant to their environment, there is no one unique way to populate this attribute. Currently Cisco has 4 hard wired formats (a-d) which are service specific and 1 configurable format (e) which can be tailored to customer and platform-specific needs.

Previously format e only allowed customizing 1 global format for all call types on a device, which limited its usefulness on devices that contained multiple services. With the extended NAS-port support, you can now configure a custom format e string for any and all service types based on the value of the NAS-Port-Type (RADIUS attribute 61). That is, when building the RADIUS Access or Accounting request, the encoding routine will pick the specific format e string defined for the session's NAS-Port-Type value and use that first instead of using the default global format e string.

The only relationship between NAS-Port-Type extensions and NAS-Port extension is that the format e string chosen by the encoding routine will depend on the value of the NAS-Port-Type for the session. Therefore if you use the extended NAS-Port-Type values (values 30-34), you should also configure format e to use them. If you do not use the extended NAS-Port-Type support, then you should use the old values, specifically, value 5 for Virtual and value 15 for Ethernet service port types. Configuring back to these port types can also allow the user to revert to previous behavior for certain interfaces.

The **radius-server attribute nas-port format e** command was enhanced to support the custom format e string with the [**type** *nas-port-type*] keyword and option. The **type** option allows you to specify different format strings to represent different physical types of ports on the Cisco 10000 for any of the extended NAS-Port-Type values. For example, you can specify the string "SSSSAAAAPPPPIIIIIIIICCCCCCCCCCCC" for type 30 (all PPPoA ports), yet you can also specify the string "SSSSAAAAPPPPVVVVVVVVVVVVVVVVVVV" for type 33 (all PPPoAoVLAN ports). In this case, the service provider can track VPI/VCI-specific information for a PPPoA user and VLAN-specific information for a PPPoEoVLAN user.

NAS-Port-ID (RADIUS Attribute 87)

The NAS-Port-ID (RADIUS attribute 87) contains the character text string identifier of the NAS port that is authenticating the user. This text string typically matches the interface description found under the CLI configuration. This attribute was previously available under Cisco Vendor Specific Attribute (VSA) "cisco-nas-port". But it is now sent by default under the IETF attribute 87 as per customer demand.

Prerequisites for Extended NAS-Port-Type and NAS-Port Attributes Support

Authentication, Authorization, and Accounting (AAA) must be enabled and already set up to use RADIUS.

Configuring Extended NAS-Port-Type and NAS-Port Attributes Support

To configure Extended NAS-Port-Type and NAS-Port Attributes Support, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server attribute 61 extended`	Enables extended, non-RFC compliant NAS-Port-Type values, which will identify new broadband service port types, such as PPPoA, PPPoEoA, PPPoEoE, PPPoEoVLAN, and PPPoEoQinQ, and sends the appropriate value to the AAA records.

Step 2	`Router(config)# radius-server attribute nas-port format e` [*string*] [**type** {*nas-port-type*}]
	
	Example:
	`Router(config)# radius-server attribute nas-port format e`
	`SSSSAPPPUUUUUUUUUUUUUUUUUUUUUUUU`
		First configure a default NAS-Port format e string that will be used as the default format by a session that has a NAS-Port-Type which is not customized for a specific service port type value.
		
		Specify a format string in configurable format e. Format e requires you to explicitly define the usage of the 32 bits of attribute 5 (NAS-Port). The usage is defined with a given parser character for each NAS-Port field of interest for a given bit field.
		
		For *string*, the characters supported are:
		• Zero : 0
		• One : 1
		• DS0 shelf : f
		• DS0 slot : s
		• DS0 adapter : a
		• DS0 port : p
		• DS0 subinterface : i
		• DS0 channel : c
		• Async shelf : F
		• Async slot : S
		• Async port : P
		• Async Line : L
		• PPPoX slot : S
		• PPPoX adapter : A
		• PPPoX port : P
		• PPPoX VLAN Id : V
		• PPPoX VPI : I
		• PPPoX VCI : C
		• Session-Id : U
		• PPPoX Inner VLAN ID: Q
		
		For more information on how to define *string*, see the *Cisco IOS Security Command Reference, Release 12.3T.*

	Command	Purpose
Step 3	`Router(config)# radius-server attribute nas-port format e` [*string*] [**type** {*nas-port-type*}] Example: `Router(config)# radius-server attribute nas-port format e SSSSAAAAPPPPIIIIIIIICCCCCCCCCCCC type 30`	Configures a specific service port type for extended NAS-Port-Type support. The **type** option allows you to specify different format strings to represent different physical types of ports on the Cisco 10000 for any of the extended NAS-Port-Type values. For example, you can specify the string "SSSSAAAAPPPPIIIIIIIICCCCCCCCCCCC" for type 30 (all PPPoA ports), yet you can also specify string "SSSSAAAAPPPPVVVVVVVVVVVVVVVVVVVV" for type 33 (all PPPoAoVLAN ports). In this case, the service provider can track VPI/VCI-specific information for a PPPoA user and VLAN-specific information for a PPPoEoVLAN user. *nas-port-type* can be one of the extended NAS-Port-Type values: • Value 30: PPPoA • Value 31: PPPoEoA • Value 32: PPPoEoE • Value 33: PPPoEoVLAN • Value 34: PPPoEoQinQ

You can override the NAS-Port-Type configured globally on the router at an interface or subinterface level. To override all global options on how the Extended NAS-Port-Type attribute is sent on any interfaces or subinterfaces such as for Ethernet, VLAN, Q-in-Q, VC, or VC ranges, enter the following commands in the PVC submode or Ethernet subinterface mode (beginning in global configuration mode):

	Command	Purpose
Step 1	`Router(config)# interface atm 5/0/0.1`	Enters ATM subinterface mode.
Step 2	`Router(config-subif)# pvc 1/33`	Enters PVC subinterface mode.
Step 3	`Router(config-if-atm-vc)# radius attribute nas-port-type` [*value*]	To set a different extended NAS-Port-Type value for an interface or subinterface, select a value for a port type to override the NAS-Port type configured globally. This feature allows for further differentiation of different customer usage. Select a *value* for NAS-Port-Type. Value can be any number, 0-2147483647, chosen by the customer. In particular, customizing your own value is useful when you need to differentiate the NAS-Port-Type based on which type of end client is actually using the port. For example if you want to track mobile clients behind a specific PVC, you can define your own NAS-Port-Type for mobile clients. **Note** You must choose a number within the range 1-40 for *value* if you wish to set a different customized extended NAS-Port-Type value and do that in conjunction with configuring an extended specific service port type (using the "format e type" command). Choosing a number outside that range will cause the default NAS-port format e string to be used to configure the NAS-Port value that is sent for that session.

Verifying Extended NAS-Port-Type and NAS-Port-ID Attributes Support

To verify the Extended NAS-Port-Type and NAS-Port-ID Attributes Support feature, enter the following command in privileged EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the current configuration of the router. Check the output of this command to confirm the configuration.

The following example displays the current configuration of RADIUS command output, where you have enabled the extended NAS-Port-Types. You can use delimiting characters to display only the relevant parts of the configuration.

```
Router# show run | inc radius
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
radius-server attribute 61 extended
radius-server attribute nas-port format e SSSSAPPPUUUUUUUUUUUUUUUUUUUUUUUUU
```

```
radius-server attribute nas-port format e SSSSAPPPIIIIIIIICCCCCCCCCCCCCCCC type 30
radius-server attribute nas-port format e SSSSAPPPIIIIIIIICCCCCCCCCCCCCCCC type 31
radius-server attribute nas-port format e SSSSAAAAPPPPVVVVVVVVVVVVVVVVVVVV type 32
radius-server attribute nas-port format e SSSSAPPPVVVVVVVVVVVVVVVVVVVVVV type 33
radius-server attribute nas-port format e SSSSAPPPQQQQQQQQQQQQQVVVVVVVVVVV type 34
radius-server host 10.76.86.91 auth-port 1645 acct-port 1646
radius-server key rad123
```

The following example displays the current configuration of RADIUS command output, where you have globally specified the format e string for all PPPoA ports (type 30).

```
Router# show run | inc radius
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
radius-server attribute nas-port format e SSSSSSSSAAAAAAAAPPPPPPPPIIIIIIII
radius-server attribute nas-port format e SSSSAAAAPPPPIIIIIIIICCCCCCCCCCCC type 30
radius-server host 10.76.86.91 auth-port 1645 acct-port 1646
radius-server key rad123
```

Configuration Examples for Extended NAS-Port-Type Attribute Support

The following examples show how to configure global support for Extended NAS-Port-Type ports, and to specify two separate e format strings globally but for two different types of ports (type 30 which is PPPoA and type 33 which is PPPoEoVLAN):

```
Router# configure terminal
Router(config)#
Router(config)# radius-server attribute 61 extended

Router(config)# radius-server attribute nas-port format e
SSSSAPPPIIIIIIIICCCCCCCCCCCCCCCC type 30
Router(config)#

Router(config)# radius-server attribute nas-port format e
SSSSAPPPVVVVVVVVVVVVVVVVVVVVVV type 33
Router(config)#
```

The following example shows you first how to customize a format e string and port type for an ATM interface and then how to override the global value set for an extended NAS-Port-Type by applying the customer-customized NAS-Port-Type value of 36 on the ATM interface:

```
Router# configure terminal
Router(config)# radius-server attribute nas-port format e SSSSAPPPIIIIIIIICCCCCCCCCCCCCCCC
type 36

Router(config)# interface atm 5/0/0.1
Router(config-subif)# pvc 1/33
Router(config-if-atm-vc)#
Router(config-if-atm-vc)# radius attribute nas-port-type 36
```

RADIUS Attribute 31: PPPoX Calling Station ID

The RADIUS Attribute 31: PPPoX Calling Station ID feature enables service providers to provide more information about the call originator to the RADIUS server in a DSL environment, such as the physical lines on which customer calls originate. Specifically, this feature allows operators to track customers through the physical lines on which customer calls originate. Service providers can better maintain the profile database of their customers as they move from one physical line to another.

Because this feature provides a virtual port that does not change as customers move from one physical line to another, RADIUS attribute 31 (Calling-Station-ID) can also be used for additional security checks. The Calling-Station-ID attribute is included in both ACCESS-REQUEST and ACCOUNTING-REQUEST messages.

The PPPoX Calling Station ID feature is described in the following topics:

Feature History for PPPoX Calling Station ID

Cisco IOS Release	Description	Required PRE
12.3(7)XI2	This feature was introduced on the Cisco 10000 series router.	PRE2

Calling-Station-ID Formats

The Calling-Station-ID attribute has 2 formats: Nas-Port and MAC-only. For Nas-Port, the system provides to the RADIUS server the host name and domain name of the node, an interface description, and VPI/VCI information (when the session is ATM-based, such as PPPoA or PPPoEoA). The MAC address is provided for PPPoEoE sessions instead of the VPI and VCI information that is provided for ATM-based sessions. For MAC-only, only the MAC address is specified for PPPoEoE sessions.

Table 16-1 summarizes the enabled Calling-Station-ID formats by session type. Notice that if both the MAC-only and Nas-Port types of Calling-Station-ID are enabled, the system provides only the MAC address to the RADIUS server for PPPoEoE sessions.

Table 16-1 Enabled Calling-Station-ID Formats by Session Type

	Enabled Calling-Station ID Format		
Session Type	**MAC-only**	**Nas-Port**	**MAC-only and Nas-Port**
PPPoA	Not applicable	hostname.domainname:int_desc:vpi:vci	hostname.domainname:int_desc:vpi:vci
PPPoEoA	Not applicable	hostname.domainname:int_desc:vpi:vci	hostname.domainname:int_desc:vpi:vci
PPPoEoE	macaddr	hostname.domainname:int_desc:macaddr	macaddr

Table 16-2 describes the Calling-Station-ID attribute fields.

Table 16-2 Calling-Station-ID Attribute Fields

Field	Description
domainname	Configured domain name of the local router
hostname	Configured host name of the local router
int_desc	Description specified for the configured ATM interface
macaddr	MAC address received from the client
vci	Virtual channel identifier (VCI) for the configured ATM interface
vpi	Virtual path identifier (VPI) for the configured ATM interface

Note The RADIUS logical line ID allows operators to download the Calling-Station-ID from RADIUS during the preauthentication phase. The RADIUS Logical Line ID feature allows a download of the attribute at session start time. You should not use the RADIUS Logical Line ID feature with the RADIUS Attribute 31: PPPoX Calling Station ID feature; using both features causes two instances of the attribute in the RADIUS IOS database for a particular user.

Restrictions for PPPoX Calling Station ID

The following restrictions apply to the RADIUS Attribute 31: PPPoX Calling Station ID feature:

- Do not use the RADIUS Logical Line ID feature with the RADIUS Attribute 31: PPPoX Calling Station ID feature. Using both features causes two instances of the attribute in the RADIUS IOS database for a particular user.
- While this feature can be used with any vendor's RADIUS server, some RADIUS servers can require modifications to their dictionary files to allow the Calling-Station-ID attribute to be presented correctly in the RADIUS logs.
- This feature supports only RADIUS; TACACS+ is not supported.
- Currently, PPPoEoVLAN and PPPoEoQinQ do not provide information on VLAN tags; only the MAC address is provided to the RADIUS server.
- RADIUS attribute 31 (Calling-Station-ID) is not supported for L2TP Network Server (LNS) environments. If you enable this attribute on an LNS, the attribute is not sent to the RADIUS server.

Related Documents for PPPoX Calling Station ID

- *RADIUS Logical Line ID* feature guide
- "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" in the *Cisco IOS Wide-Area Networking Configuration Guide, Release 12.2*
- *Cisco IOS Dial Technologies Configuration Guide, Release 12.2*

Configuration Tasks for PPPoX Calling Station ID

To configure the RADIUS Attribute 31: PPPoX Calling Station ID feature, perform the following configuration tasks:

- Configuring the Calling-Station-ID Format
- Verifying the Calling-Station-ID

Configuring the Calling-Station-ID Format

To configure the Calling-Station-ID format, perform the following task in global configuration mode:

To verify the Extended NAS-Port-Type and NAS-Port-ID Attributes Support feature, enter the following command in privileged EXEC mode:

Command	Purpose
`Router(config)# radius-server attribute 31 pppox` *format*	*format*—Specifies the type of Calling-Station-ID • **nas-port**—Enables the Nas-Port format of the Calling-Station-ID attribute. • **mac-addr**—Enables the Mac-only format of the Calling-Station-ID attribute. You can enable one or both types of Calling-Station ID. If both the MAC-only and Nas-Port types of Calling-Station-ID are enabled, the system provides only the MAC address to the RADIUS server for PPPoEoE sessions. See Table 16-1 on page 16-14 for a summary of enabled Calling-Station-ID formats by session type.

Verifying the Calling-Station-ID

To verify the Calling-Station-ID, perform the following task in EXEC mode use the **debug radius** command in privileged EXEC mode. The **debug radius** command verifies that RADIUS attribute 31, Calling-Station-ID, is in the ACCESS-REQUEST and ACCOUNTING-REQUEST. Example 16-2 shows sample output of the **debug radius** command.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use

debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Example 16-2 debug radius Command Output

```
*Sep 14 14:54:43.259: RADIUS(00000008): Send Access-Request to 10.0.0.8:1645 id1645/34,
len 121
*Sep 14 14:54:43.259: RADIUS: authenticator C3 81 6B 7A F8 38 F9 FE - E6 82 A6 91 92 54 44
66
*Sep 14 14:54:43.259: RADIUS: Framed-Protocol [7] 6 PPP [1]
*Sep 14 14:54:43.259: RADIUS: User-Name [1] 8 "johndoe"
*Sep 14 14:54:43.259: RADIUS: CHAP-Password [3] 19 *
*Sep 14 14:54:43.259: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
*Sep 14 14:54:43.259: RADIUS: NAS-Port [5] 6 0
*Sep 14 14:54:43.259: RADIUS: NAS-Port-Id [87] 9 "8/0/0/0"
*Sep 14 14:54:43.259: RADIUS: Calling-Station-Id [31] 35
":c10k.xtnet.com:my_interface:00b0.c2ef.8400"
*Sep 14 14:54:43.259: RADIUS: Service-Type [6] 6 Framed [2]
*Sep 14 14:54:43.259: RADIUS: NAS-IP-Address [4] 6 10.0.0.119
```

Configuration Example for PPPoX Calling Station ID

The following PPP termination aggregation (PTA) and L2TP access concentrator (LAC) example shows how to configure your LAC for preauthorization by downloading the Logical Line ID:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa session-id common
!
hostname c10k
ip domain-name xtnet.com
!-------- hostname and domain name are included in the nas-port type CSID---

vc-class atm ppp_auto1200
  vpn service service_control
  protocol pppoe group PPPOETEST
  encapsulation aal5autoppp Virtual-Template1 group PPPOETEST
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Loopback10
 ip address 172.16.1.1 255.255.255.0
!
interface FastEthernet0/0/0
 ip address 10.0.0.119 255.255.255.0
 speed 100
 full-duplex
!
interface ATM1/0/0
 no ip address
 shutdown
 no atm pxf queuing
 atm ilmi-keepalive
 pvc 0/16 ilmi
!
!
```

```
interface ATM1/0/1
 no ip address
 atm clock INTERNAL
 no atm auto-configuration
 atm ilmi-keepalive
 no atm address-registration
 pvc 0/16 ilmi
!
!
interface ATM1/0/1.111 multipoint
! -----This description is used in the calling-station-id ------
 description test_descr
 pvc 0/100
  class-vc ppp_auto1200
!
 pvc 0/101
  class-vc ppp_auto1200
!
interface GigabitEthernet8/0/0
 ip address 10.10.0.1 255.255.255.0
 negotiation auto
 pppoe enable group PPPOETEST
!
interface Virtual-Template1
 ip unnumbered Loopback0
 peer default ip address pool default
 ppp authentication chap callin
!
ip local pool default 3.3.3.1 3.3.3.10
!
radius-server attribute 31 pppox nas-port
radius-server attribute 31 pppox mac-addr
radius-server attribute 32 include-in-access-req
radius-server host 10.0.0.8 auth-port 1645 acct-port 1646 key cisco
```

Related Commands for PPPoX Calling Station ID

Command	Description
ip radius source-interface	Requires RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets

RADIUS Packet of Disconnect

The RADIUS Packet of Disconnect feature consists of a method for terminating a session that has already been connected. This packet of disconnect (POD) is a RADIUS access_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect the user after the session has been accepted by the RADIUS access_accept packet. This may be needed in at least two situations:

- Detection of fraudulent use, which cannot be performed before accepting the session. A price structure so complex that the maximum session duration cannot be estimated before accepting the session. This may be the case when certain types of discounts are applied or when multiple users use the same subscription simultaneously.

- To prevent unauthorized servers from disconnecting users, the authorizing agent that issues the POD packet must include three parameters in its packet of disconnect request. For a session to be disconnected, all parameters must match their expected values at the router. If the parameters do not match, the router discards the packet of disconnect packet and sends a NACK (negative acknowledgement message) to the agent.

The data parameters are the following RADIUS attributes:

- User- Name (RADIUS IETF attribute 1)
- Framed-IP-Address (RADIUS IETF attribute 8)
- Acct-Session-Id (RADIUS IETF attribute 44)
- Session-Svr-Key (vendor-proprietary RADIUS attribute 151)

For information about RADIUS attributes, see Appendix A, "RADIUS Attributes".

The RADIUS Packet of Disconnect feature is discussed in the following topics:

Feature History for RADIUS Packet of Disconnect

Cisco IOS Release	Description	Required PRE
12.3(7)XI1	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.	PRE2

Benefits for RADIUS Packet of Disconnect

- Ability to terminate an established session

Restrictions for RADIUS Packet of Disconnect

Proper matching identification information must be communicated by the:

- Billing server and router configuration
- Router's original accounting start request
- Server's POD request

Related Documents for RADIUS Packet of Disconnect

- *Cisco IOS Security Configuration Guide, Release 12.2*
- *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2*
- *Cisco Access Registrar 3.5 Installation and Configuration Guide*
- RFC 2865, *Remote Authentication Dial-in User Service*

Prerequisites for RADIUS Packet of Disconnect

- Configure AAA as described in *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.2.

Configuration Tasks for RADIUS Packet of Disconnect

To configure the RADIUS Packet of Disconnect feature, perform the following configuration tasks:

- Configuring AAA POD Server
- Verifying AAA POD Server

Configuring AAA POD Server

To configure the Calling-Station-ID format, perform the following task in global configuration mode:

Command	Purpose
`Router(config)# aaa pod server clients [`*client ip address*`] port` [*port-number*] [**auth-type** {**any**| **all**| **session-key**}] [ignore {session-key | server-key}] **server-key** *string*	Enables inbound user sessions to be disconnected when specific session attributes are presented. *client ip-address*—(Optional) Registers the IP address of all the clients who can send POD requests. If not set, it can receive a POD request from any client. *port-number*—(Optional) The network access server User Datagram Protocol (UDP) port to use for POD requests. Default value is 1700. **auth-type—**(Optional) The type of authorization required for disconnecting sessions. • **any**—Session that matches all of the attributes sent in the POD packet is disconnected. The POD packet may contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key). • **all**—Only a session that matches all four key attributes is disconnected. **All** is the default. • **session-key**—Session with a matching session-key attribute is disconnected. All other attributes are ignored. **ignore—**(Optional) Ignore the session key or the server key received in the POD packet for session matching. **server-key—**Configures the shared-secret text string. • *string*—The shared-secret text string that is shared between the network access server and the client workstation. This shared-secret string must be the same on both systems.

Verifying AAA POD Server

To verify that the router is configured correctly to performs an AAA POD server, enter the **show running-configuration** command in privileged EXEC mode to display the command settings for the router.

```
Router# show running-configuration
!
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
aaa pod server clients <ip address> port <port number> auth-type [all/ any/ session-key]
server-key cisco
```

Monitoring and Maintaining AAA POD Server

To monitor an AAA POD server and troubleshoot problems:

- Ensure that the POD port is configured correctly in both the router (using **aaa pod server** command) and the RADIUS server. Both should be the same.
- Ensure that the shared-secret key configured in the router (using **aaa pod server** command) and in the AAA server are the same.
- Use debug commands:
  - **debug aaa pod**—displays debug messages for POD packets
  - **debug aaa authentication**—displays debug messages for authentication
  - **debug aaa accounting**—displays debug messages for accounting records
  - **debug radius**—displays debug messages for RADIUS packets

  The following example shows output from the **debug aaa pod** command and indicates a successful POD request.

```
Router# debug aaa pod
AAA POD packet processing debugging is on
General OS:
AAA POD packet processing debugging is on

Router#
4d18h:  ++++++ POD Attribute List ++++++
4d18h: 6291C598 0 00000009 username(336) 8 pod_user
4d18h: 7085EE1C 0 00000001 nas-ip-address(439) 4 23.3.7.3
4d18h:
4d18h: POD: 2.0.0.210 user pod_user 0.0.0.0 sessid 0x0 key 0x0
4d18h: POD:      Line     User     IDB          Session Id Key
4d18h: POD: Skip          <NULL>   0.0.0.0      0x363      0x0
4d18h: POD: KILL Virtual- pod_user 104.1.2.38   0x421A     0xD4105397
4d18h: POD: Skip Virtual- <NULL>   0.0.0.0      0x421B     0x0
4d18h: POD: Sending ACK from port 3799 to 2.0.0.210/64917
```

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use debug commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use.

Configuration Example for RADIUS Packet of Disconnect

Example 16-3 provides a configuration example for a router performing as an AAA POD server:

Example 16-3 Configuring a Router as an AAA POD Server

```
Router(config)# aaa pod server server-key xyz123
```

CHAPTER 17

Configuring L2 Virtual Private Networks

To improve profitability, service providers (SPs) introduce new services to reduce operational expenditures. To reduce the number of managed networks, use network convergence, a multiphase transition of the network. This affects both the core and edge/aggregation side. The technology is predominantly Multiprotocol Label Switching (MPLS) based core networks. However, IP cores are the service of choice in a number of large SPs. Both the IP and the MPLS cores carry multiservice traffic. The edges of the network is constructed with network elements providing a single network element for convergence between Layer 2 and Layer 3 services.

The following Layer 2 virtual private network (L2VPN) solutions enable existing or emerging Layer 2 transport technology to interwork through converged MPLS or IP core networks.

- Virtual Private Wire Services (VPWS)—A point-to-point service consisting of individual point-to-point connections cross-connected to native interfaces.
- Virtual Private LAN Services (VPLS)—A service consisting of a set of point-to-multipoint connections.

L2VPN features are of the VPWS type and are designed for the benefit of the carriers. L2VPN features allow for a transparent use of network resources, and a way of reducing the number of networks that need managing.

Cisco nonstop forwarding (NSF) with stateful switchover (SSO) is effective at increasing availability of network services. Cisco NSF with SSO provides continuous packet forwarding, even during a network processor hardware or software failure. In a redundant system, the secondary processor recovers control plane service during a critical failure in the primary processor. SSO synchronizes network state information between the primary and the secondary processor."

Any Transport over MPLS (AToM) uses NSF, SSO, and Graceful Restart to allow a route processor (RP) to recover from a disruption in control plane service without losing the MPLS forwarding state. In Cisco IOS Release 12.2(33) SB, the L2VPN features support NSF/SSO. See the "NSF and SSO—L2VPN" section on page 17-6.

Cisco 10000 series routers also support the following two L2VPN technology solutions:

- Local Switching (LS)—The ordered duple <AC, AC>. This is the point-to-point interconnection of two attachment circuits within a Cisco 10000 series router chassis. Also, two attachment circuits (ACs) can be of:
  - The same type—Creating a like-to-like LS connection.
  - A distinct type—Creating an any-to-any LS connection.
- AToM—The ordered triple <AC, PW, AC>. This is the point-to-point interconnection of two attachment circuits in separate Cisco 10000 series router chassis through a pseudowire (MPLS). Also, two ACs can be of the same type in which case a like-to-like AToM connection exists. Or, two ACs can be of a distinct type, in which case an any-to-any AToM connection exists.

Using the Label Distribution Protocol (LDP), an AToM circuit session is identified by a unique VC (virtual circuit) between two PE routers. When a Layer 2 frame is received by the imposition PE router, it is encapsulated in an MPLS packet with a VC label, IGP label, and possibly other labels. When the MPLS packet reaches the disposition PE router, the packet is converted back into its Layer 2 encapsulation.

AToM encapsulates Layer 2 frames at the ingress (or imposition) provider edge (PE) router, and sends them to a corresponding PE router at the other end of the connection. The corresponding router is the egress (or disposition) PE router, and it removes the encapsulation and sends out the Layer 2 frame.

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a *pseudowire*, between the routers. An AToM circuit is one type of pseudowire connection.

Benefits of Enabling Layer 2 Packets to Send in an MPLS Network

Some of the benefits of enabling Layer 2 packets to be sent in the MPLS network include:

- The AToM product set accommodates many types of Layer 2 packets, including Ethernet and Frame Relay, across multiple Cisco router platforms. This enables the service provider to transport all types of traffic over the backbone and accommodate all types of customers.
- AToM adheres to the standards developed for transporting Layer 2 packets over MPLS. (See the "Standards and RFCs" section on page 17-5 for the specific standards that AToM follows.) This benefits the service provider who wants to incorporate industry-standard methodologies in the network. Other Layer 2 solutions are proprietary, which can limit the service provider's ability to expand the network and can force the service provider to use only one vendor's equipment.
- Upgrading to AToM is transparent to the customer. Because the service provider network is separate from the customer network, the service provider can upgrade to AToM without disruption of service to the customer. The customers assume that they are using a traditional Layer 2 backbone.

A control word (also referred to as a *shim* header) can be added at the imposition router and, if so, this control word is removed at the disposition router.

Cisco 10000 series router supports up to 8000 attachment circuits (ACs). An AToM circuit use one AC and a LS circuit use two ACs. Therefore, Cisco 10000 series router supports 8000 AToM connections or 4000 LS connections or any combination of both AToM and LS connections that sums up to 8000 ACs. Also, Tunnel selection allows you to specify the path that AToM traffic uses. See the "Any Transport over MPLS—Tunnel Selection" section on page 17-47.

This chapter contains the following topics:

- Any Transport over MPLS—Tunnel Selection, page 17-47

Feature History for L2VPN

Cisco IOS Release	Description	Required PRE
12.2(28)SB	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(31)SB2	Support was added for the PRE3.	PRE3
12.2(31)SB2	Ethernet to VLAN over AToM (Bridged) functionality was added.	PRE2/PRE3
12.2(33)SB	The following L2VPN features were added on the Cisco 10000 series router: • IEEE 802.1Q Tunneling (QinQ) for AToM • NSF/SSO - Any Transport over MPLS (AToM) • Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown • Any Transport over MPLS: Tunnel Selection • L2VPN Local Switching - HDLC/PPP • Ethernet/VLAN to ATM AAL5 Interworking • Ethernet VLAN to Frame Relay Interworking	PRE2/PRE3/PRE4

Supported L2VPN Transport Types

In Cisco IOS Release 12.2(28)SB, the Cisco 10000 series router supports the following AToM transport types:

- ATM AAL5 SDU support over MPLS
- Ethernet over MPLS
  - VLAN mode
  - Port mode
- Frame Relay over MPLS
  - DLCI-to-DLCI connections
  - Port-to-port connections
- HDLC over MPLS
- PPP over MPLS

Note Functionally, both HDLC over MPLS and Frame Relay port-to-port connections are the same.

Prerequisites for L2VPN: AToM

Before configuring L2VPN, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other using IP.
- Configure the label distribution protocol to be Label Distribution Protocol (LDP).
- Configure label-switched paths (LSPs) between the PE routers. To enable dynamic MPLS labeling on all paths between the imposition and disposition PE routers, use the **mpls ip** command.
- Configure a loopback interface for originating and terminating Layer 2 traffic. Make sure the PE routers can access the other router's loopback interface. Note that the loopback interface is not needed in all cases. For example, tunnel selection does not need a loopback interface when AToM is directly mapped to a TE tunnel.

Note For L2VPN: LS, it is *not* necessary to configure:

—The label distribution protocol to be Label Distribution Protocol (LDP).

—Label-switched paths (LSPs) between the PE routers using the **mpls ip** command.

Supported Line Cards

Table 17-1 lists line cards supported by the Cisco 10000 series router.

Table 17-1 Cisco 10000 Series Line Cards that Support L2VPN

Transport Type	Supported Line Cards
ATM AAL5 SDU support over MPLS	4-port OC-3/STM-1 ATM 8-port E3/DS3 ATM 1-port OC-12 ATM
Ethernet over MPLS: VLAN mode Port mode	8-port Fast Ethernet Half-Height 1-port Gigabit Ethernet Half-Height 1-port Gigabit Ethernet SIP-600 SPA-1X10GE-L-V2 (10GE) SPA-2X1GE-V2 (2 port GE) SPA-5X1GE-V2 (5 port GE)
Frame Relay over MPLS: DLCI-to-DLCI connections Port-to-port connections HDLC over MPLS PPP over MPLS	24-port Channelized E1/T1 1-port Channelized OC-12/STM-4 4-port Channelized OC-3/STM-1 4-port Channelized T3 6-port Channelized T3 8-port Unchannelized E3/T3 6-port OC-3/STM1 Packet over SONET 1-port OC-12 Packet over SONET 1-port OC-48/STM-16 Packet over SONET

Restrictions for L2VPN

The L2VPN feature has the following restrictions:

- Address format: Configure the LDP router ID on all PE routers to be a loopback address with a /32 mask. Otherwise, some configurations might not function properly.
- The size of maximum transmission unit (MTU) must be the same at both ends of the circuit. To avoid fragmentation of the packets along the way, ensure that the size of MTU at both end of the circuit is smaller than the size of MTU in the core.
- The following L2VPN features are *not* supported:
  - ATM cell switching of any kind
  - ATM AAL5 PDU mode
  - Fragmentation and reassembly, as defined in "PWE3 Fragmentation and Reassembly," draft-ietf-pwe3-fragmentation-05.txt, February 2004
  - Sequence number support in the control word
  - Tunnel stitching
  - Pseudowire termination

Standards and RFCs

L2VPN conforms to the industry standards and RFCs listed in Table 17-2.

Table 17-2 Standards and RFCs Supported by L2VPN

Standard or RFC	Title
draft-martini-l2circuit-trans-mpls-08.txt	*Transport of Layer 2 Frames over MPLS*
draft-martini-l2circuit-encap-mpls-04.txt	*Encapsulation Methods for Transport of Layer 2 Frames over MPLS*
RFC 3032	*MPLS Label Stack Encoding*
RFC 3036	*LDP Specification*

MIBs

Table 17-3 lists the MIBs that L2VPN supports.

Table 17-3 MIBs Supported by L2VPN

Transport Type	MIB
ATM AAL5 SDU support over MPLS	MPLS LDP MIB (MPLS-LDP-MIB.my) ATM MIB (ATM-MIB.my) CISCO AAL5 MIB (CISCO-AAL5-MIB.my) Cisco Enterprise ATM Extension MIB (CISCO-ATM-EXT-MIB.my) Supplemental ATM Management Objects (CISCO-IETF-ATM2-PVCTRAP-MIB.my) Interfaces MIB (IF-MIB.my)
Ethernet over MPLS: VLAN mode Port mode	CISCO-ETHERLIKE-CAPABILITIES.my Ethernet MIB (ETHERLIKE-MIB.my) Interfaces MIB (IF-MIB.my) MPLS LDP MIB (MPLS-LDP-MIB.my)
Frame Relay over MPLS: DLCI-to-DLCI connections Port-to-port connections	Cisco Frame Relay MIB (CISCO-FRAME-RELAY-MIB.my) Interfaces MIB (IF-MIB.my) MPLS LDP MIB (MPLS-LDP-MIB.my)
HDLC over MPLS	MPLS LDP MIB (MPLS-LDP-MIB.my)
PPP over MPLS	Interface MIB (IF-MIB.my)

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at:

http://tools.cisco.com/go/mibs

NSF and SSO—L2VPN

L2VPN NSF improves the availability of a service provider's network that uses AToM to provide Layer 2 VPN services to its customers. High availability (HA) provides the ability to detect failures and manage them with minimal disruption to the service being provided. L2VPN NSF is achieved by SSO and NSF mechanisms. A standby RP provides control-plane redundancy. The control plane state and data plane provisioning information for the attachment circuits (ACs) and AToM pseudowires (PWs) are checkpointed to the standby RP to provide NSF for AToM L2VPNs.

Checkpointing AToM Information

Checkpointing is a function that copies state information from the active RP to the backup RP, thereby ensuring that the backup RP has the latest information. If the active RP fails, the backup RP can take over.

For the L2VPN NSF feature, the checkpointing function copies the active RP's information bindings to the backup RP. The active RP sends updates to the backup RP when information is modified.

To display checkpointing data, issue the **show acircuit checkpoint** command on the active and backup RPs. The active and backup RPs have identical copies of the information.

Checkpointing Troubleshooting Tips

To help troubleshoot checkpointing errors, enter the following commands:

- **debug acircuit checkpoint** command—To enable checkpointing debug messages for ACs.
- **debug mpls l2transport checkpoint** command—To enable checkpointing debug messages for AToM.
- **show acircuit checkpoint** command—To display the AC checkpoint information.
- **show mpls l2transport checkpoint** command—To display if checkpointing is allowed, the quantity of AToM VCs that were bulk-synced (on the active RP), and the quantity of AToM VCs that have checkpoint data (on the standby RP).
- **show mpls l2transport vc detail** command—To display details of VC checkpointed information.

The NSF/SSO - L2VPN feature is described in the following topics:

- Prerequisites for NSF/SSO - L2VPN, page 17-7
- Restrictions for NSF/SSO - L2VPN, page 17-8
- Configuring NSF/SSO - L2VPN, page 17-8
- Configuration Examples of NSF/SSO—Layer 2 VPN, page 17-9

Prerequisites for NSF/SSO - L2VPN

This section lists the following prerequisites for the feature:

- Neighbor Routers in the MPLS HA Environment
- Stateful Switchover
- Nonstop Forwarding for Routing Protocols

Neighbor Routers in the MPLS HA Environment

Cisco 10000 routers must be used as the neighboring device.

Stateful Switchover

For information on this topic, see the *Stateful Switchover* section in the *NSF/SSO: Any Transport over MPLS and Graceful Restart* document at:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsatomha.html#wp1098167

Nonstop Forwarding for Routing Protocols

For information on this topic, see the *Nonstop Forwarding for Routing Protocols* section in the *NSF/SSO: Any Transport over MPLS and Graceful Restart* document at:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsatomha.html#wp1098561

Restrictions for NSF/SSO - L2VPN

For information on this topic, see the *Restrictions for AToM NSF* section in the *NSF/SSO: Any Transport over MPLS and Graceful Restart* document at:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsatomha.html#wp1068923

Configuring NSF/SSO - L2VPN

For information on this topic, see the *How to Configure AToM NSF* section in the *NSF/SSO: Any Transport over MPLS and Graceful Restart* document at:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsatomha.html#wp1112888

Configuration Examples of NSF/SSO—Layer 2 VPN

Example 17-1 illustrates how to configure AToM NSF on two PE routers:

Example 17-1 Ethernet to VLAN Interworking with AToM NSF

PE1	PE2
ip cef ! redundancy mode sso ! mpls ldp graceful-restart mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 force mpls ldp advertise-tags ! pseudowire-class atom-eth encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/1/0 xconnect 10.9.9.9 123 encap mpls pw-class atom_eth interface POS6/1/0 ip address 10.1.1.1 255.255.255.0 mpls ip mpls label protocol ldp clock source internal crc 32 ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 no shutdown ! router ospf 10 nsf ietf network 10.8.8.8 0.0.0.0 area 0 network 19.1.1.1 0.0.0.0 area 0	ip cef ! redundancy mode sso ! mpls ldp graceful-restart mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 force mpls ldp advertise-tags ! pseudowire-class atom-eth encapsulation mpls interworking eth ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet3/0/0 ip route-cache cef ! interface FastEthernet3/0/0.3 encapsulation dot1Q 10 xconnect 10.8.8.8 123 encap mpls pw-class atom_eth interface POS1/0/0 ip address 10.1.1.2 255.255.255.0 mpls ip mpls label protocol ldp clock source internal crc 32 ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! router ospf 10 nsf ietf network 10.9.9.9 0.0.0.0 area 0 network 10.1.1.2 0.0.0.0 area 0

Note NSF must be enabled for routing protocols. You can use either the **cisco** or **ietf** option. Example 17-1 has the **ietf** option because it is a standard option, whereas **cisco** is proprietary option.

L2VPN Local Switching—HDLC/PPP

The L2VPN Local Switching - HDLC/PPP feature enables service providers to support different encapsulations over HDLC local switched circuits that function as back-to-back circuits. The provisioned HDLC Local Switched circuits can also be backed by using PWRED.

Prerequisites of L2VPN Local Switching—HDLC/PPP

In Cisco IOS Release 12.2(33)SB, the L2VPN Local Switching - HDLC/PPP, you must ensure that interfaces must be HDLC encapsulated on the PE router. The CE routers can choose any HDLC-based encapsulation, including Frame Relay and PPP.

Restrictions of L2VPN Local Switching—HDLC/PPP

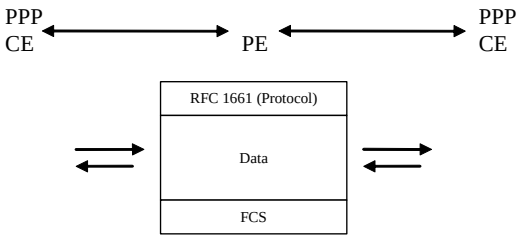
In Cisco IOS Release 12.2(33)SB, the L2VPN Local Switching - HDLC/PPP feature has the following restrictions:

- On the PE HDLC interface, the IP address cannot be configured because it conflicts with the **connect** command.
- Interworking is not supported on HDLC/PPP interfaces.
- Only same-speed interfaces should be connected, to avoid arbitrary packet drops due to a higher speed interface overrunning a lower speed one.
- For some HDLC/PPP applications which are sensitive to time delay, the PE may introduce some network delay, enough to prevent the HDLC/PPP link from coming up because of a protocol timeout (an ISDN Q921 link).

PPP Like-to-Like Local Switching

Some applications, such as transport of compressed voice between the two CEs, require a setup of an end-to-end PPP session between two CE routers that are connected to the same PE router. In such cases, HDLC pass-through mechanism is proposed and the interworking scenario is simplified to PPP transport for like-to-like services. PPP local switching functionality on the PE router provides simple HDLC connectivity between two end-users found on different CE routers as shown in Figure 17-1.

Figure 17-1 PPP Local Switching



The interfaces are HDLC encapsulated on the PE router. The CE routers may use PPP-based encapsulation.

Frames manipulated by the PE router preserve the PPP header as described in RFC-1661.

HDLC Like-to-Like Local Switching

Like PPP, HDLC sessions can be forwarded between two CE routers connected to the same PE router. The microcode implements a HDLC pass-through mechanism for the HDLC traffic. As the service provided is equivalent to a back-to-back serial connection between the two CE routers, the connection should be between same-speed interfaces with the matched Maximum Transmission Unit (MTU) configuration. There are no QoS requirements on the PE router since one interface cannot overrun another.

The interfaces are HDLC encapsulated on the PE router. CE routers may use any HDLC-based encapsulation, including Frame Relay.

Configuration Tasks and Examples

You can configure the L2VPN Local Switching - HDLC/PPP feature on a PE router using the following steps:

1. **config t**
2. **interface serial** slot/subslot/port:channel-id
3. **encapsulation** hdlc
4. **interface serial** slot/subslot/port:channel-id
5. **encapsulation** hdlc
6. **connect** connection-name interface interface

The following example shows you how to configure the L2VPN Local Switching - HDLC/PPP feature on the PE router:

```
config t
interface serial 3/0/20:0
 encapsulation hdlc
interface serial 4/0/11:9
 encapsulation hdlc
 connect hdlcls serial3/0/20:0 serial4/0/11:9
```

Note Because the default encapsulation of a serial interface is HDLC, the **encapsulation** command is optional. However, when you configure the CE router, you must specify the **encapsulation** command because of the difference in configuration.

You can configure PPP on the CE router using the following steps:

1. **config t**
2. **interface serial** slot/subslot/port:channel-id
3. **encapsulation** ppp

You can configure HDLC on the CE router using the following steps:

1. **config t**
2. **interface serial** slot/subslot/port:channel-id
3. **encapsulation** hdlc

Configuration Tasks for L2VPN

To configure L2VPN, you have to configure the following L2VPN features:

Setting Up the Pseudowire—AToM Circuit

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of a connection called a pseudowire between the routers. You specify the following information on *each* PE router:

- The type of Layer 2 data to be transported across the pseudowire, such as Ethernet, Frame Relay, or ATM
- The IP address of the loopback interface of the peer PE router, which enables the PE routers to communicate
- A unique combination of peer PE IP address and VC ID that identifies the pseudowire

To set up a pseudowire connection or AToM circuit between two PE routers, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# pseudowire-class name`	(Optional) Establishes a pseudowire class with a name that you specify and specifies the tunneling encapsulation. It is not necessary to specify a pseudowire class if you specify the tunneling method as part of the **xconnect** command. The pseudowire-class configuration group specifies the characteristics of the tunneling mechanism, including: • Encapsulation type • Control protocol • Payload-specific options
Step 2	`Router(config)# interface interface-type interface-number`	Defines the interface or subinterface on the PE router.
Step 3	`Router(config-if)# encapsulation encapsulation-type`	Specifies the encapsulation type for the interface, such as dot1q.
Step 4	`Router(config-if)# xconnect peer-router-id vcid encapsulation mpls`	Makes a connection to the peer PE router by specifying the LDP router ID of the peer PE router. Identifies a unique identifier that is shared between the two PE routers. The *vcid* is a 32-bit identifier. **Note** The combination of the peer-router-id and the VC ID must be a unique combination on the router. Two circuits cannot use the same combination of peer-router-id and VC ID. Specifies the tunneling method used to encapsulate data in the pseudowire. For AToM, the tunneling method used to encapsulate data is **mpls**.

Example 17-2 shows a sample configuration for the ATM AAL5 SDU over MPLS transport. The PVC on 0/100 is configured for AAL5 transport.

Example 17-2 ATM AAL5 SDU Support over MPLS

```
interface ATM4/0
    pvc 0/100 l2transport
        encapsulation aal5
        xconnect 13.13.13.13 100 encapsulation mpls
```

Configuring ATM AAL5 SDU Support over MPLS

ATM AAL5 SDU support over MPLS encapsulates ATM AAL5 service data units (SDUs) in MPLS packets and forwards them across the MPLS network. Each ATM AAL5 SDU is transported as one packet.

To configure ATM AAL5 SDU support over MPLS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 2	`Router(config-if)# pvc [name] vpi/vci l2transport`	Creates or assigns a name to an ATM PVC. The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC. Enters L2transport VC configuration mode.
Step 3	`Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. Make sure you specify the same encapsulation type on the PE and CE routers.
Step 4	`Router(config-if-atm-l2trans-pvc)# xconnect peer-router-id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.

Example 17-3 shows how to enable ATM AAL5 SDU support over MPLS on an ATM PVC.

Example 17-3 ATM AAL5 SDU Support over MPLS on an ATM PVC

```
interface atm1/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 13.13.13.13 100 encapsulation mpls
```

Verifying ATM AAL5 SDU Support over MPLS

To verify that ATM AAL5 SDU support over MPLS is configured on a PVC, issue the **show mpls l2transport vc** command. Example 17-4 shows sample output for this command.

Example 17-4 show mpls l2transport vc Command Output

```
Router# show mpls l2transport vc

Local intf   Local circuit          Dest address      VC ID      Status
---------    -------------          ------------      -----      ------
ATM1/0       ATM AAL5 1/100         4.4.4.4           100        UP
```

Configuring ATM-to-ATM PVC Local Switching

The following ATM line cards are supported for Cisco 10000 series routers:

- 4-port OC-3/STM-1
- 8-port E3/DS3

- 1-port OC-12

To configure ATM-to-ATM PVC local switching, enter the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/port`	Specifies an ATM interface and enters interface configuration mode.
Step 2	`Router(config-if)# pvc vpi/vci l2transport`	Assigns a virtual path identifier (VPI) and virtual channel identifier (VCI). The **l2transport** keyword indicates that the permanent virtual circuit (PVC) is a switched PVC instead of a terminated PVC.
Step 3	`Router(cfg-if-atm-l2trans-pvc)# encapsulation layer-type`	Specifies the encapsulation type for the PVCs, AAL5 is the only layer type supported. Repeat Steps 1, 2, and 3 for another ATM PVC on the same router.
Step 4	`Router(config)# connect connection-name interface pvc interface pvc`	Creates a local connection between the two specified PVCs.

Example 17-5 shows how to enable ATM AAL5 SDU mode Layer 2 local switching.

Example 17-5 Enabling ATM AAL5 SDU Mode Layer 2 Local Switching

```
interface atm 1/0/0
   pvc 0/100 l2transport
     encapsulation aal5

interface atm 2/0/0
   pvc 0/50 l2transport
     encapsulation aal5

connect conn1 atm 1/0/0 0/100 atm 2/0/0 0/50
```

Configuring OAM Cell Emulation for ATM AAL5 SDU Support over MPLS

If a PE router does not support the transport of Operation, Administration, and Maintenance (OAM) cells across an LSP, you can use OAM cell emulation to locally terminate or loop back the OAM cells. You configure OAM cell emulation on both PE routers, which emulates a VC by forming two unidirectional LSPs. You use the **oam-ac emulation-enable** and **oam-pvc manage** commands on both PE routers to enable OAM cell emulation.

After you enable OAM cell emulation on a router, you can configure and manage the ATM VC in the same manner as you would a terminated VC. A VC that is configured with OAM cell emulation can send loopback cells at configured intervals toward the local CE router.

The endpoint can be either of the following:

- End-to-end loopback, which sends OAM cells to the local CE router.
- Segment loopback, which responds to OAM cells to a device along the path between the PE and CE routers.

The OAM cells include the following:

- Alarm indication signal (AIS)
- Remote defect indication (RDI)

These cells identify and report defects along a VC. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream devices affected by the failure. When a router receives an AIS cell, it marks the ATM VC down and sends an RDI cell to let the remote end know about the failure.

Note For AAL5 SDU support over MPLS, you can configure the **oam-pvc manage** command only after you issue the **oam-ac emulation-enable** command.

You can configure OAM cell emulation for ATM AAL5 SDU support over MPLS in the following ways:

- Configuring OAM Cell Emulation for ATM AAL5 SDU Support over MPLS on PVCs, page 17-16
- Configuring OAM Cell Emulation for ATM AAL5 SDU Support over MPLS in VC Class Configuration Mode, page 17-18

Configuring OAM Cell Emulation for ATM AAL5 SDU Support over MPLS on PVCs

To configure OAM cell emulation for ATM AAL5 SDU support over MPLS on a PVC, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 2	`Router(config-if)# pvc [name] vpi/vci l2transport`	Creates or assigns a name to an ATM PVC. The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC. Enters L2 Transport VC configuration mode.
Step 3	`Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. Make sure you specify the same encapsulation type on the PE and CE routers.
Step 4	`Router(config-if-atm-l2trans-pvc)# xconnect peer-router-id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 5	`Router(config-if-atm-l2trans-pvc)# oam-ac emulation-enable [ais-rate]`	Enables OAM cell emulation for AAL5 over MPLS. The *ais-rate* variable lets you specify the rate at which AIS cells are sent. The range is 0 to 60 seconds. The default is 1 second, which means that one AIS cell is sent every second.
Step 6	`Router(config-if-atm-l2trans-pvc)# oam-pvc manage [frequency]`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. The optional *frequency* variable is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.

Example 17-6 shows how to enable OAM cell emulation on an ATM PVC.

Example 17-6 OAM Cell Emulation on an ATM PVC

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 13.13.13.13 100 encapsulation mpls
oam-ac emulation-enable
oam-pvc manage
```

Example 17-7 shows how to set the rate at which an AIS cell is sent to every 30 seconds.

Example 17-7 Setting the AIS Send Rate in OAM Cell Emulation on an ATM PVC

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 13.13.13.13 100 encapsulation mpls
oam-ac emulation-enable 30
oam-pvc manage
```

Verifying OAM Cell Emulation on an ATM PVC

In Example 17-8, the **show atm pvc** command shows that OAM cell emulation is enabled on the ATM PVC.

Example 17-8 show atm pvc Command Output

```
Router# show atm pvc 5/500

ATM4/1/0.200: VCD: 6, VPI: 5, VCI: 500
UBR, PeakRate: 1
AAL5-LLC/SNAP, etype:0x0, Flags: 0x34000C20, VCmode: 0x0
OAM Cell Emulation: enabled, F5 End2end AIS Xmit frequency: 1 second(s)
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not ManagedVerified
ILMI VC state: Not Managed
InPkts: 564, OutPkts: 560, InBytes: 19792, OutBytes: 19680
InPRoc: 0, OutPRoc: 0
InFast: 4, OutFast: 0, InAS: 560, OutAS: 560
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
Out CLP=1 Pkts: 0
OAM cells received: 26
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 26
OAM cells sent: 77
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutAIS: 77, F5 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Configuring OAM Cell Emulation for ATM AAL5 SDU Support over MPLS in VC Class Configuration Mode

The following steps explain how to configure OAM cell emulation as part of a VC class. You can then apply the VC class to an interface, a subinterface, or a VC. When you configure OAM cell emulation in VC class configuration mode and then apply the VC class to an interface, the settings in the VC class apply to all the VCs on the interface, unless you specify a different OAM cell emulation value at a lower level, such as the subinterface or VC level.

For example, you can create a VC class that specifies OAM cell emulation and sets the rate of AIS cells to every 30 seconds. You can apply the VC class to an interface. Then, for one PVC, you can enable OAM cell emulation and set the rate of AIS cells to every 15 seconds. All the PVCs on the interface use the cell rate of 30 seconds, except for the one PVC that was set to 15 seconds.

To enable OAM cell emulation as part of a VC class and apply it to an interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vc-class atm name`	Creates a VC class and enters VC class configuration mode.
Step 2	`Router(config-vc-class)# encapsulation layer-type`	Configures the ATM adaptation layer (AAL) and encapsulation type.
Step 3	`Router(config-vc-class)# oam-ac emulation-enable [ais-rate]`	Enables OAM cell emulation for AAL5 over MPLS. The *ais-rate* variable lets you specify the rate at which AIS cells are sent. The range is 0 to 60 seconds. The default is 1 second, which means that one AIS cell is sent every second.
Step 4	`Router(config-vc-class)# oam-pvc manage [frequency]`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. The optional *frequency* variable is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.
Step 5	`Router(config-vc-class)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# interface type slot/port`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 7	`Router(config-if)# class-int vc-class-name`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	`Router(config-if)# pvc [name] vpi/vci l2transport`	Creates or assigns a name to an ATM PVC. The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC. Enters L2 Transport VC configuration mode.
Step 9	`Router(config-if-atm-l2trans-pvc)# xconnect peer-router-id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.

Example 17-9 configures OAM cell emulation for ATM AAL5 SDU support over MPLS in VC class configuration mode. The VC class is then applied to an interface.

Example 17-9 OAM Cell Emulation for ATM AAL5 SDU Support over MPLS in VC Class Configuration Mode—VC Class Applied to an Interface

```
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
class-int oamclass
pvc 1/200 l2transport
xconnect 13.13.13.13 100 encapsulation mpls
```

Example 17-10 shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

Example 17-10 OAM Cell Emulation for ATM AAL5 SDU Support over MPLS in VC Class Configuration Mode—VC Class Applied to a PVC

```
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
pvc 1/200 l2transport
class-vc oamclass
xconnect 13.13.13.13 100 encapsulation mpls
```

Example 17-11 shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface. One PVC is configured with OAM cell emulation at an AIS rate of 10. That PVC uses the AIS rate of 10 instead of 30.

Example 17-11 OAM Cell Emulation for ATM AAL5 SDU Support over MPLS in VC Class Configuration Mode—VC Class Applied to an Interface

```
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0
class-int oamclass
pvc 1/200 l2transport
oam-ac emulation-enable 10
xconnect 13.13.13.13 100 encapsulation mpls
```

Configuring Ethernet over MPLS

Ethernet over MPLS works by encapsulating Ethernet protocol data units (PDUs) in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet. Several methods exists for configuring Ethernet over MPLS:

- VLAN mode—Transports Ethernet traffic from a source 802.1Q VLAN to a destination 802.1Q VLAN over a core MPLS network.

- Port mode—Allows a frame coming into an interface to be packed into an MPLS packet and transported over the MPLS backbone to an egress interface. The entire Ethernet frame is transported without the preamble or FCS as a single packet.
- VLAN ID Rewrite—Enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel.

You can configure Ethernet over MPLS in the following ways:

- Configuring Ethernet over MPLS in VLAN Mode, page 17-20
- Configuring Ethernet over MPLS in Port Mode, page 17-21
- Configuring Ethernet over MPLS with VLAN ID Rewrite, page 17-27

Ethernet over MPLS Restrictions

The following restrictions pertain to the Ethernet over MPLS transport:

- Packet format: Ethernet over MPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames. The Inter-Switch Link (ISL) protocol is not supported between the PE and customer edge (CE) routers.
- When the first Ethernet over MPLS in VLAN mode circuit is configured, the controller (the entire port) is automatically placed in promiscuous mode. The promiscuous mode is removed only when the last Ethernet over MPLS in VLAN mode circuit associated with that controller is removed.
- The AToM control word is supported. However, if the peer PE router does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.

Configuring Ethernet over MPLS in VLAN Mode

A virtual LAN (VLAN) is a switched network that is logically segmented by functions, project teams, or applications regardless of the physical location of users. Ethernet over MPLS allows you to connect two VLAN networks that are in different locations. You configure the PE routers at each end of the MPLS backbone and add a point-to-point virtual circuit (VC). Only the two PE routers at the ingress and egress points of the MPLS backbone know about the VCs dedicated to transporting Layer 2 VLAN traffic. All other routers do not have table entries for those VCs.

For Ethernet over MPLS in VLAN mode, it is possible for VPN circuits to coexist with pseudowire circuits. Because the port is in promiscuous mode, the frames are filtered by the VLAN ID.

Note You must configure Ethernet over MPLS in VLAN mode on the subinterfaces. However, you cannot configure Ethernet over MPLS (VLAN mode) on a Q-in-Q subinterface.

To configure Ethernet over MPLS in VLAN mode, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/interface.subinterface`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 2	`Router(config-subif)# encapsulation dot1q vlan-id`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not need to be on the same subnet.
Step 3	`Router(config-subif)# xconnect peer-router-id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

Configuring Ethernet over MPLS in Port Mode

Port mode allows a frame coming into an interface to be packed into an MPLS packet and transported over the MPLS backbone to an egress interface. The entire Ethernet frame without the preamble or FCS is transported as one packet. To configure port mode, use the **xconnect** command in main interface mode and specify the destination address and the VC ID. The syntax and semantics of the **xconnect** command are the same as for all other transport types. Each interface is associated with one unique pseudowire VC label.

When configuring Ethernet over MPLS in port mode, use the following guidelines:

- The pseudowire VC type is set to Ethernet.
- Port mode and Ethernet VLAN mode are mutually exclusive. If you enable a main interface for port-to-port transport, you cannot also enter commands on a subinterface.

To configure Ethernet over MPLS in port mode, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/interface`	Specifies the Gigabit Ethernet interface and enters interface configuration mode. Make sure the interface on the adjoining CE router is on the same VLAN as this PE router.
Step 2	`Router(config-if)# xconnect peer-router-id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

Example 17-12 shows how to configure VC 123 in Ethernet port mode:

Example 17-12 Ethernet over MPLS in Port Mode

```
pseudowire-class ethernet-port
  encapsulation mpls

interface gigabitethernet1/0
  xconnect 10.0.0.1 123 pw-class ethernet-port
```

Note Depending on the interface type, you can also use the **interface fastethernet** command.

Verifying Ethernet over MPLS in VLAN Mode and Port Mode

To determine if a VC is set up in VLAN mode or port mode, issue the **show mpls l2transport vc** command.

Example 17-13 shows two VCs set up for Ethernet over MPLS:

- VC 2 is set up in Ethernet VLAN mode.
- VC 8 is set up in Ethernet port mode.

Example 17-13 show mpls l2transport vc Command Output

```
Router# show mpls l2transport vc

Local intf     Local circuit         Dest address    VC ID      Status
-------------  --------------------- --------------- ---------- ----------
Gi4/0.1        Eth VLAN 2            11.1.1.1        2          UP
Gi8/0/1        Ethernet              11.1.1.1        8          UP
```

If you issue the **show mpls l2transport vc detail** command, the output is similar, as shown in Example 17-14.

Example 17-14 show mpls l2transport vc detail Command Output

```
Router# show mpls l2transport vc detail

Local interface: Gi4/0.1 up, line protocol up, Eth VLAN 2 up
Destination address: 11.1.1.1, VC ID: 2, VC status: up

...
Local interface: Gi8/0/1 up, line protocol up, Ethernet up
  Destination address: 11.1.1.1, VC ID: 8, VC status: up
```

IEEE 802.1Q Tunneling for AToM—QinQ

The IEEE 802.1Q Tunneling (QinQ) for AToM feature is described in the following topics:

Prerequisites for IEEE 802.1Q Tunneling (QinQ) for AToM

In Cisco IOS software Release 12.2(33)SB, the QinQ (short for 802.1Q-in-802.1Q) tunneling and tag rewrite feature is supported on the following Cisco 10000 series engines and line cards:

- PRE-2, PRE-3, and PRE-4 engines
- 8-port Fast Ethernet line card (ESR-HH-8FE-TX)
- 2-port half-height Gigabit Ethernet line card (ESR-HH-1GE)
- 1-port full-height Gigabit Ethernet line card (ESR-1GE)
- SPA line cards

Restrictions for IEEE 802.1Q Tunneling (QinQ) for AToM

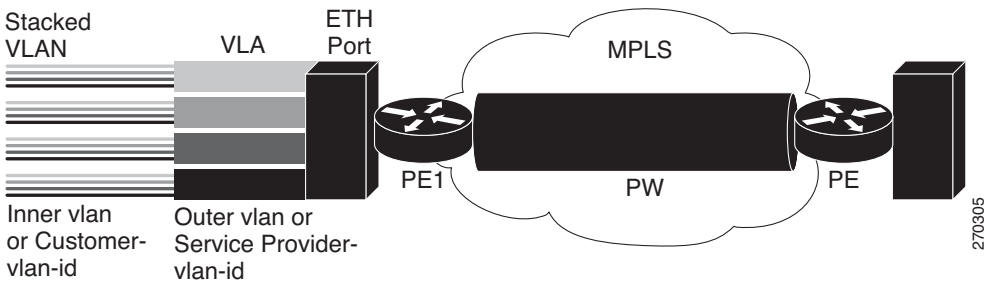
In Cisco IOS Release 12.2(33)SB, the QinQ tunneling and tag rewrite feature has the following restrictions:

- Up to a maximum of 447 outer-VLAN IDs and up to 4095 inner VLAN IDs can be supported for the Ethernet QinQ over AToM feature.
- Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. i.e. The Ethernet VLAN QinQ rewrite of both VLAN Tags capability is supported only on ethernet sub-interfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

Ethernet VLAN Q-in-Q AToM

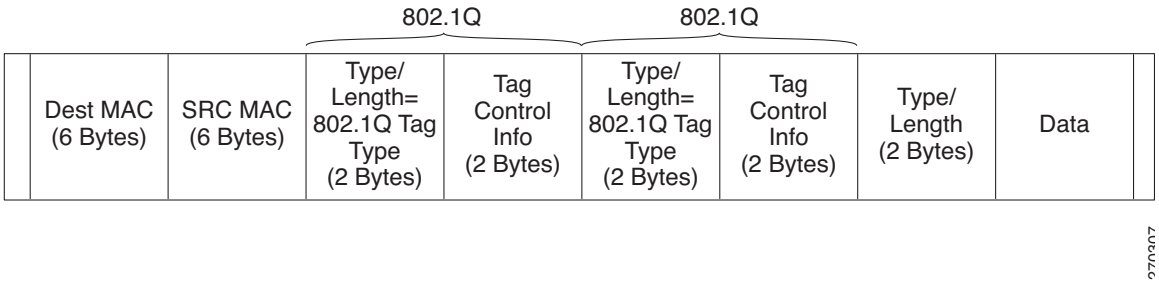
In Metro Ethernet deployment, in which CE routers and PE routers are connected through an Ethernet switched access network, packets that arrive at PE routers can contain up to two IEEE 802.1q VLAN tags (one inner VLAN tag which identifies the customer; and another outer VLAN tag which denotes the customer's service provider). This technique of allowing multiple VLAN tagging on the same Ethernet packet and creating a stack of VLAN IDs is known as QinQ (short for 802.1Q-in-802.1Q). Figure 17-2 shows how different edge devices can do L2 switching on the different levels of the VLAN stack.

Figure 17-2 Ethernet VLAN QinQ



When the outer VLAN tag is the service-delimiting VLAN tag, QinQ packets are processed similar to the ones with one VLAN tag (case previously named Ethernet VLAN Q-in-Q modified, which is already supported in the 12.2(31) SB release). However, when a customer must use a combination of the outer and inner VLAN tags to delimit service for customers, the edge device should be able to choose a unique pseudowire based on a combination of the inner and outer VLAN IDs on the packet shown in Figure 17-3. The customer may want to be able to rewrite both the inner and the outer VLAN IDs on the traffic egress side.

Figure 17-3 Ethernet VLAN QinQ Header



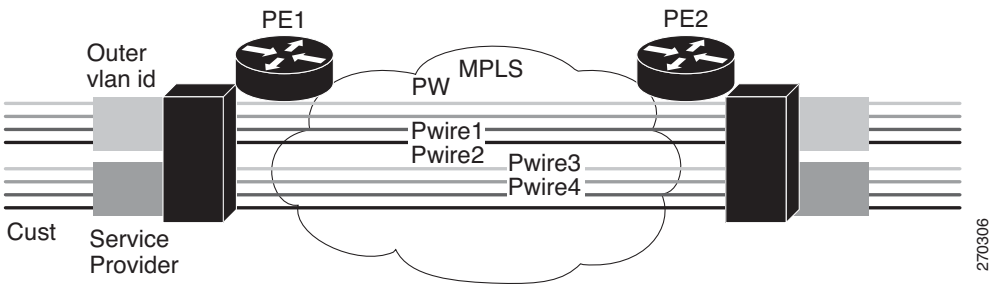
The IEEE 802.1Q Tunneling (QinQ) for AToM can be further explained as follows:

- QinQ Tunneling Based on Inner and Outer VLAN Tags, page 17-24
- Rewriting Inner and Outer VLAN Tags on QinQ Frames, page 17-24

QinQ Tunneling Based on Inner and Outer VLAN Tags

When handling incoming QinQ Ethernet traffic, the Cisco 10000 series edge router allows a customer to choose a unique pseudowire endpoint to switch the traffic based on the combination of inner and outer VLAN IDs. For example, Figure 17-4 shows how a unique pseudowire is selected depending upon the combination of inner (customer edge) and outer (service provider) VLAN IDs. Thus, traffic for different customers can be kept separate.

Figure 17-4 QinQ Connection



Rewriting Inner and Outer VLAN Tags on QinQ Frames

When managing incoming AToM Ethernet QinQ traffic, the Cisco 10000 edge router:

1. Strips off the MPLS labels.
2. Allows the customer to rewrite both the inner and outer VLAN IDs before sending the packets to the egress QinQ interface. Note this capability is provided only for AToM like-to-like Ethernet QinQ traffic.

Support for these features is added in Cisco IOS Release 12.2(33). The QinQ AToM feature is a like-to-like interworking case over AToM. This feature requires changes to the microcode to allow it to overwrite two layers of VLAN tags on Ethernet QinQ traffic, transported across AToM pseudowires.

- On the ingress side—The packets preserve their L2 header with the two VLAN tags, and it is sent across the pseudowire with VC type of 4.

- On the egress side—The MPLS label is stripped, and up to two levels of VLAN tags are rewritten per the configuration.

Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. The Ethernet VLAN Q-in-Q rewrite of both VLAN Tags capability is supported only on Ethernet sub-interfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

Configuration Examples

Example 17-15 shows an unambiguous QinQ configured on GigE subinterface.

Example 17-15 Unambiguous QinQ

```
interface GigabitEthernet1/0/0.100
encapsulation dot1q 100 second-dot1q 200
xconnect 23.0.0.16 410 encapsulation mpls
```

Example 17-16 shows an ambiguous QinQ configured on a GigE subinterface.

Example 17-16 Ambiguous QinQ

```
interface GigabitEthernet1/0/0.200
encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000
xconnect 23.0.0.16 420 encapsulation mpls
interface GigabitEthernet1/0/0.201
encapsulation dot1q 201 second-dot1q any
xconnect 23.0.0.16 430 encapsulation mpls
```

Note Ambiguous inner VLAN IDs are not supported in this release.

Verifying QinQ AToM

Example 17-17 shows the command output of the **show mpls l2transport vc** command, which is used to verify the VC set up in EoMPLS QinQ mode.

Example 17-17 show mpls l2transport vc Command Output

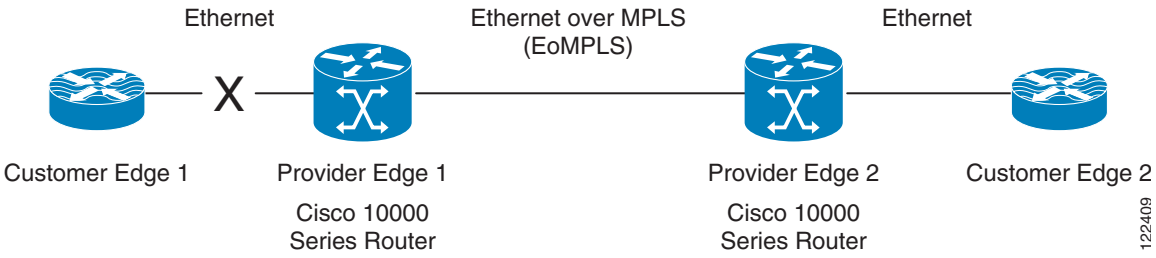
```
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi1/0/0.1      Eth VLAN:100/200           100.1.1.2       1          UP
```

Remote Ethernet Port Shutdown

This Cisco IOS feature allows a service provider edge (PE) router on the local end of an Ethernet over MPLS (EoMPLS) pseudowire to detect a remote link failure and shutdown of the Ethernet port on the local customer edge (CE) router. Because the Ethernet port on the local CE router is shut down, the router does not lose data by continuously sending traffic to the failed remote link. This is beneficial if the link is configured as a static IP route.

Figure 17-5 illustrates a condition in an EoMPLS wide area network (WAN) with a down Layer 2 tunnel link between a CE1 router and the PE1 router. A CE2 router on the far side of the Layer 2 tunnel, continues to forward traffic to CE1 through the L2 tunnel.

Figure 17-5 Remote Link Outage in EoMPLS Wide Area Network



In earlier releases than Cisco IOS Release 12.2(33)SB, the PE2 router did not detect a failed remote link. Traffic forwarded from CE2 to CE1 is lost until routing or spanning tree protocols detected the down remote link. If the link was configured with static routing, remote link outage can be difficult to detect by the L3 routing protocol.

With Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown, the PE2 router detects the remote link failure and causes a shutdown of the local CE2 Ethernet port. When the remote L2 tunnel link is restored, the local interface is automatically restored as well. The possibility of data loss is thus diminished.

With reference to Figure 17-5, a remote Ethernet shutdown sequence occurs as follows:

1. The remote link between CE1 and PE1 fails.
2. PE2 with remote Ethernet port shutdown enabled detects the remote link failure and disables the transmit laser on the line card interface connected to CE2.
3. CE2 receives an RX_LOS error alarm causing CE2 to bring down the interface.
4. PE2 maintains its interface with CE2 in an up state.
5. When the remote link and EoMPLS connection is restored, the PE2 router enables the transmit laser.
6. The CE2 router brings up its downed interface.

Restrictions for Configuring Remote Ethernet Port Shutdown

The following restrictions pertain to the Remote Ethernet Port Shutdown feature:

- For Cisco IOS Release 12.2(33) SB, this feature is implemented for port mode Ethernet over MPLS connections between Cisco 10000 series Ethernet line cards only.
- This feature is not symmetrical if the remote PE router is running an older version of image or is on another platform that does not support the EoMPLS remote Ethernet port shutdown feature and the local PE is running an image which supports this feature.

Configuring Remote Ethernet Port Shutdown

By default, the Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown feature is automatically enabled when an image with this feature supported is loaded on the Cisco 10000 series router. However, to enable the Remote Ethernet Port Shutdown feature, enter the **remote link failure notification** command, as shown in Example 17-18.

To disable the feature, enter the **no remote link failure notification** command (Example 17-19).

Example 17-18 Enabling Remote Ethernet Port Shutdown under the Xconnect Configuration

```
pseudowire-class eompls
```

```
 encapsulation mpls
!
interface GigabitEthernet1/0/0
 xconnect 1.1.1.1 1 pw-class eompls
  remote link failure notification
!
```

Example 17-19 Disabling Remote Ethernet Port Shutdown under the Xconnect Configuration

```
pseudowire-class eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
 xconnect 1.1.1.1 1 pw-class eompls
  no remote link failure notification
!
```

To see the operational status of all remote L2 tunnels by interface, enter **show interface** and **show ip interface brief** commands, as shown in Example 17-20.

Example 17-20 Operational Status for All Remote L2 Tunnels by Interface

```
router# show interface GigabitEthernet1/0/0
GigabitEthernet1/0/0 is L2 Tunnel remote down, line protocol is up
  Hardware is Half-height Gigabit Ethernet MAC Controller, address is 0009.b68f.9b18 (bia
0009.b68f.9b18)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
  ......
  ......

router# sh ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0/0      24.3.8.1        YES NVRAM  up                    up
GigabitEthernet1/0/0   unassigned      YES NVRAM  L2 Tunnel remote down up
GigabitEthernet2/0/0   30.1.1.1        YES manual up                    up
```

Enter **show controller** and **show controller** *interface* commands to see the port transceiver state, as shown in Example 17-21.

Example 17-21 Port Transceiver State

```
router# show controller GigabitEthernet1/0/0
Interface GigabitEthernet1/0/0(idb 0x4FB5CA7C)
Hardware is Half-height Gigabit Ethernet MAC Controller, network connection mode is auto
  network link is L2 Tunnel remote down
  loopback type is none
  ......
```

Configuring Ethernet over MPLS with VLAN ID Rewrite

The VLAN ID Rewrite feature enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel. The Cisco 10000 series router automatically performs VLAN ID Rewrite on the disposition PE router. There is no configuration required.

Configuring Frame Relay over MPLS

Frame Relay over MPLS encapsulates Frame Relay protocol data units (PDUs) in MPLS packets and forwards them across the MPLS network. For Frame Relay, you can set up data-link connection identifier (DLCI)-to-DLCI connections or port-to-port connections.

- With DLCI-to-DLCI connections, the PE routers manipulate the packet by removing headers, adding labels, and copying control word elements from the header to the PDU.
- With port-to-port connections, you use HDLC mode to transport the Frame Relay encapsulated packets. In HDLC mode, the entire HDLC packet is transported. Only the HDLC flags and FCS bits are removed. The contents of the packet are not used or changed, including the FECN, BECN, and DE bits.

Note Frame Relay traffic shaping is not supported with AToM-switched VCs.

You can configure Frame Relay over MPLS in the following ways:

- Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections, page 17-28
- Configuring Frame Relay over MPLS with Port-to-Port Connections, page 17-29
- Enabling Other PE Devices to Transport Frame Relay Packets, page 17-30

Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections

To configure Frame Relay over MPLS with DLCI-to-DLCI connections, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# frame-relay switching`	Enables permanent virtual circuit (PVC) switching on a Frame Relay device.
Step 2	`Router(config)# interface serial` *slot/port*	Specifies a serial interface and enters interface configuration mode.
Step 3	`Router(config-if)# encapsulation frame-relay [cisco | ietf]`	Specifies Frame Relay encapsulation for the interface. You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 4	`Router(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. You can also specify the interface to support NNI and DTE connections.
Step 5	`Router(config-if)# exit`	Exits from interface configuration mode.

	Command	Purpose
Step 6	`Router(config)# connect connection-name interface dlci l2transport`	Defines connections between Frame Relay PVCs and enters connect submode. Using the **l2transport** keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network. The *connection-name* argument is a text string that you provide. The *interface* argument is the interface on which a PVC connection is defined. The *dlci* argument is the DLCI number of the PVC that will be connected.
Step 7	`Router(config-fr-pw-switching)# xconnect peer-router-id vcid encapsulation mpls`	Creates the VC to transport the Layer 2 packets. In a DLCI-to DLCI connection type, Frame Relay over MPLS uses the **xconnect** command in connect submode.

Example 17-22 shows how to enable Frame Relay over MPLS with DLCI-to-DLCI connections.

Example 17-22 Frame Relay over MPSL with DLCI-to-DLCI Connections

```
frame-relay switching
interface Serial3/1
encapsulation frame-relay ietf
frame-relay intf-type dce
exit
connect fr1 Serial 5/0 1000 l2transport
xconnect 10.0.0.1 123 encapsulation mpls
```

Configuring Frame Relay over MPLS with Port-to-Port Connections

When you set up a port-to-port connection between PE routers, you use HDLC mode to transport the Frame Relay encapsulated packets. Perform this task to set up Frame Relay port-to-port connections.

To configure Frame Relay over MPLS with port-to-port connections, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/port`	Specifies a serial interface and enters interface configuration mode.
Step 2	`Router(config-if)# encapsulation hdlc`	Specifies that Frame Relay PDUs is encapsulated in HDLC packets.
Step 3	`Router(config-if)# xconnect peer-router-id vcid encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Example 17-23 shows how to enable Frame Relay over MPLS with port-to-port connections.

Example 17-23 Frame Relay over MPLS With Port-to-Port Connections

```
interface serial5/0
encapsulation hdlc
xconnect 10.0.0.1 123 encapsulation mpls
```

Enabling Other PE Devices to Transport Frame Relay Packets

You can configure an interface as a data terminal equipment (DTE) device or a data circuit-terminating equipment (DCE) switch, or as a switch connected to a switch with network-to-network interface (NNI) connections. Use the following command in interface configuration mode:

frame-relay intf-type [dce | dte | nni]

The following table explains the keywords:

Keyword	Description
dce	Enables the router or access server to function as a switch connected to a router.
dte	Enables the router or access server to function as a DTE device. DTE is the default.
nni	Enables the router or access server to function as a switch connected to a switch.

Local Management Interface and Frame Relay over MPLS

Local Management Interface (LMI) is a protocol that communicates status information about permanent virtual circuits (PVCs). When a PVC is added, deleted, or changed, the LMI notifies the endpoint of the status change. LMI also provides a polling mechanism that verifies that a link is up.

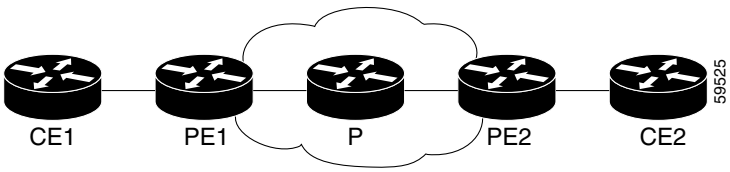
LMI Process

To determine the PVC status, LMI checks that a PVC is available from the reporting device to the Frame Relay end-user device. If a PVC is available, LMI reports that the status is "Active," which means that all interfaces, line protocols, and core segments are operational between the reporting device and the Frame Relay end-user device. If any of those components is not available, the LMI reports a status of "Inactive."

Note Only the DCE and NNI interface types can report LMI status.

Figure 17-6 is a sample topology that illustrates how LMI works.

Figure 17-6 Sample LMI Topology



In Figure 17-6, note the following:

- CE1 and PE1 and PE2 and CE2 are Frame Relay LMI peers.
- CE1 and CE2 can be Frame Relay switches or end-user devices.
- Each Frame Relay PVC is composed of multiple segments.
- The DLCI value is local to each segment and is changed as traffic is switched from segment to segment. Two Frame Relay PVC segments exist in Figure 17-6; one is between PE1 and CE1 and the other is between PE2 and CE2.

The LMI protocol behavior depends on DLCI-to-DLCI connections versus port-to-port connections.

DLCI-to-DLCI Connections

If DLCI-to-DLCI connections are configured, LMI runs locally on the Frame Relay ports between the PE and CE devices.

- CE1 sends an active status to PE1 if the PVC for CE1 is available. If CE1 is a switch, LMI checks that the PVC is available from CE1 to the user device attached to CE1.
- PE1 sends an active status to CE1 if the following conditions are met:
  - A PVC for PE1 is available.
  - PE1 received an MPLS label from the remote PE router.
  - An MPLS tunnel label exists between PE1 and the remote PE router.

For DTE/DCE configurations, the following LMI behavior exists:

The Frame Relay device accessing the network (DTE) does not report PVC status. Only the network device (DCE) or NNI can report status. Therefore, if a problem exists on the DTE side, the DCE is not aware of the problem.

Port-to-Port Connections

If port-to-port connections are configured, the PE routers do not participate in the LMI status-checking procedures. LMI operates between the CE routers only. The CE routers must be configured as DCE-DTE or NNI-NNI.

For information about LMI, including configuration instructions, see the "Configuring the LMI" section of the *Configuring Frame Relay* document.

Configuring Frame Relay-to-Frame Relay Local Switching

Frame Relay switching is a means of switching packets based upon the data link connection identifier (DLCI), which can be looked upon as the Frame Relay equivalent of a MAC address. You perform the switching by configuring your router or access server as a Frame Relay network. There are two parts to a Frame Relay network: the Frame Relay data terminal equipment (DTE) (the router or access server) and the Frame Relay data communications equipment (DCE) switch.

Local switching allows you to switch Layer 2 data between two interfaces of the same type for example, ATM-to-ATM, or Frame-Relay-to-Frame-Relay.

For background information about Frame-Relay-to-Frame-Relay Local Switching, see the *Distributed Frame Relay Switching* feature guide.

You can switch between virtual circuits on the same port, as detailed in the "Configuring Frame Relay Same-Port Switching" section on page 17-33.

The following channelized line cards are supported for the Cisco 10000 series routers:

- 1-port channelized OC-12/STM-4
- 4-port channelized OC-3/STM-1
- 6-port channelized T3
- 24-port channelized E1/T1

The following packet over SONET line cards are supported for the Cisco 10000 series routers:

- 1-port OC-12 Packet over SONET
- 1-port OC-48/STM-16 Packet over SONET
- 6-port OC-3/STM-1 Packet over SONET

The Frame Relay-to-Frame Relay Local Switching feature is described in the following topics:

- Configuring Frame Relay for Local Switching, page 17-32
- Configuring Frame Relay Same-Port Switching, page 17-33
- Verifying Layer 2 Local Switching for Frame Relay, page 17-34
- Configuring QoS Features, page 17-34

Configuring Frame Relay for Local Switching

To configure Frame Relay for local switching, enter the following commands, beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# frame-relay switching`	Enables Permanent Virtual Circuits (PVCs) switching on a Frame Relay DCE device or a Network-to-Network Interface (NNI).
Step 2	`Router(config)# interface type number`	Specifies an interface and enters interface configuration mode.
Step 3	`Router(config-if)# encapsulation frame-relay [cisco | ietf]`	Enables Frame Relay encapsulation. • **cisco**—Cisco's own encapsulation (default) • **ietf**—Internet Engineering Task Force (IETF) standard (RFC 1490). Use this keyword when connecting to another vendor's equipment across a Frame Relay network.
Step 4	`Router(config-if)# frame-relay interface-dlci dlci switched`	(Optional) Creates a switched PVC and enters Frame Relay DLCI configuration mode. Repeat Step 1 through Step 4 for each switched PVC. If you do not create a Frame Relay PVC in this step, it is automatically created in Step 6 by the **connect** command.
Step 5	`Router(config-fr-dlci)# exit`	Exits Frame Relay DLCI configuration mode and returns to global configuration mode.
Step 6	`Router(config)# connect connection-name interface dlci interface dlci`	Defines a connection between Frame Relay PVCs.

Example 17-24 configures Frame-Relay-to-Frame-Relay for local switching.

Example 17-24 Configuring Frame Relay-to-Frame Relay for Local Switching

```
frame-relay switching
interface serial 1/0/0.1/1:0
encapsulation frame-relay
frame-relay interface-dlci 100 switched
exit
connect connection1 serial1/0/0.1/1:0 100 serial2/0/0.1/2:0 101
```

Configuring Frame Relay Same-Port Switching

Use the following steps to configure local Frame Relay same-port switching on a single interface, beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay DCE device or a NNI.
Step 2	`Router(config)# interface type number`	Specifies the interface and enters interface configuration mode.
Step 3	`Router(config-if)# encapsulation frame-relay [cisco | ietf]`	Enables Frame Relay encapsulation. • **cisco**—Cisco's own encapsulation (default) • **ietf**—Internet Engineering Task Force (IETF) standard (RFC 1490). Use this keyword when connecting to another vendor's equipment across a Frame Relay network.
Step 4	`Router(config-if)# frame-relay intf-type {dce | dte | nni}`	(Optional) Enables support for a particular type of connection. • **dce**—data communications equipment • **dte**—data terminal equipment • **nni**—network-to-network interface
Step 5	`Router(config-if)# frame-relay interface-dlci dlci switched`	(Optional) Creates a switched PVC and enters Frame Relay DLCI configuration mode. Repeat Step 1 through Step 5 for each switched PVC. If you do not create a Frame Relay PVC in this step, it is automatically created in Step 8 by the **connect** command.
Step 6	`Router(config-fr-dlci)# exit`	Exits Frame Relay DLCI configuration mode and returns to interface configuration mode.
Step 7	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	`Router(config)# connect connection-name interface dlci interface dlci`	Defines a connection between the two data links.

Example 17-25 shows how to configure Frame Relay same-port switching.

Example 17-25 Configuring Frame Relay Same-Port Switching

```
frame-relay switching
interface serial 1/0/0.1/1:0
encapsulation frame-relay
frame-relay intf-type nni
frame-relay interface-dlci 100 switched
exit
exit
connect connection1 serial1/0 100 serial1/0 200
```

Verifying Layer 2 Local Switching for Frame Relay

To verify configuration of the Layer 2 Local Switching feature, use the **show connection frame-relay-to-frame-relay** command and the **show frame-relay pvc** command in privileged EXEC mode.

Example 17-26 shows the output of the **show connection frame-relay-to-frame-relay** command, which displays the local connection between a Frame Relay interface and a Frame Relay local switching interface.

Example 17-26 show connection frame-relay-to-frame-relay Command Output

```
Router# show connection frame-relay-to-frame-relay
ID  Name            Segment 1              Segment 2           State
=================================================================
1   fr2fr           Se3/0/0.1/1:0 100       Se3/0/0.1/2:0 200  UP
```

Example 17-27 shows the output of the **show frame-relay pvc** command, which shows a switched Frame Relay PVC.

Example 17-27 show frame-relay pvc Command Output

```
Router# show frame-relay pvc 16
PVC Statistics for interface POS5/0 (Frame Relay NNI)
DLCI = 16, DLCI USAGE = SWITCHED, PVC STATUS = UP, INTERFACE = POS5/0
LOCAL PVC STATUS = UP, NNI PVC STATUS = ACTIVE
input pkts 0 output pkts 0 in bytes 0
out bytes 0 dropped pkts 100 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 0 out bcast bytes 0
switched pkts 0
Detailed packet drop counters:
no out intf 0 out intf down 100 no out PVC 0
in PVC down 0 out PVC down 0 pkt too big 0
pvc create time 00:25:32, last time pvc status changed 00:06:31
```

Configuring QoS Features

For information about configuring QoS features on the Cisco 10000 series router, see the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Table 17-4 and Table 17-5 outline the level of support for modular QoS CLI (MQC) commands as they relate to Frame Relay DLCI interfaces.

The values shown in the tables are as follows:

- No—You cannot perform this policy map action
- Yes—You can perform this policy map action
- N/A (not applicable)—You can apply the policy map action but it does not have any effect on packets

Table 17-4 Frame Relay DLCI Input Policy Map Actions

Policy Map Actions	Frame Relay DLCI Interface
bandwidth	no
queue-limit	no
priority	no
shape	no
random-detect	no
set ip prec/dscp	N/A
set qos-group	yes
set discard class	yes
set atm-clp	N/A
set fr-de	no
set cos	no
police	yes
set mpls-exp topmost	N/A
set mpls-exp imposition	N/A

Table 17-5 Frame Relay Output (Disposition Router) Policy Map Actions

Policy Map Actions	Frame Relay DLCI Interface
bandwidth	yes
queue-limit	yes
priority	yes
shape	yes
random-detect	yes (discard class only)
set ip prec/dscp	N/A
set qos-group	N/A
set discard class	yes
set atm-clp	no
set fr-de	not supported
set cos	no
police	yes
set mpls-exp topmost	N/A

Configuring HDLC and PPP over MPLS

With HDLC over MPLS, the entire HDLC packet is transported. The ingress PE router removes only the HDLC flags and frame check sequence (FCS) bits. The contents of the packet are not used or changed.

With PPP over MPLS, the ingress PE router removes the flags, address, control field, and the FCS.

HDLC over MPLS is described in:

Restrictions for HDLC over MPLS

The following restrictions pertain to the HDLC over MPLS feature:

- Asynchronous interfaces: Asynchronous interfaces are not supported.
- Interface configuration: You must configure HDLC over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

Restrictions for PPP over MPLS

The following restrictions pertain to the PPP over MPLS feature:

- Asynchronous interfaces—Are not supported. The connections between the CE and PE routers on both ends of the backbone must have similar link layer characteristics. The connections between the CE and PE routers must both be synchronous.
- Multilink PPP (MLP)—Is not supported.
- Interface configuration: You must configure PPP on router interfaces only. You cannot configure PPP on subinterfaces.

Configuring HDLC over MPLS or PPP over MPLS

To configure HDLC over MPLS or PPP over MPLS, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/port`	Specifies a serial interface and enters interface configuration mode. You must configure HDLC and PPP over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.
Step 2	`Router(config-if)# encapsulation encapsulation-type`	Specifies HDLC or PPP encapsulation and enters connect submode. *encapsulation-type* can be HDLC or PPP.
Step 3	`Router(config-fr-pw-switching)# xconnect peer-router-id vcid encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Estimating the Size of Packets Traveling Through the Core Network

The following calculation helps you determine the size of the packets traveling through the core network. You set the maximum transmission unit (MTU) on the core-facing interfaces of the P and PE routers to accommodate packets of this size.

The MTU should be greater than or equal to the total bytes of the items in the following equation:

```
Core MTU >= (Edge MTU + Transport header + AToM header + (MPLS label stack size* MPLS
label size))
```

The following sections describe the variables used in the equation.

Edge MTU

The edge MTU is the MTU for the customer-facing interfaces.

Transport Header

The transport header depends on the transport type. Table 17-6 lists the specific sizes of the headers.

Table 17-6 Header Size of Packets

Transport Type	Header Size (bytes)
ATM AAL5 SDU support over MPLS	12
Ethernet over MPLS in VLAN mode	18
Ethernet over MPLS in port mode	14
Frame Relay over MPLS with DLCI-to-DLCI connections	4
HDLC over MPLS	4
PPP over MPLS	4

AToM Header

The AToM header is 4 bytes (control word). The Cisco 10000 series router adds the control word for all supported transport types by default.

MPLS Label Stack

The MPLS label stack size depends on the configuration of the core MPLS network.

- AToM uses one MPLS label to identify the AToM VCs (VC label). Therefore, the minimum MPLS label stack is 1 for directly connected AToM PE routers, which are PE routers that do not have a P router between them.
- If LDP is used in the MPLS network, the label stack size is 2 (the LDP label and the VC label).
- If a TE tunnel instead of LDP is used between PE routers in the MPLS network, the label stack size is 2 (the TE label and the VC label).
- If a TE tunnel and LDP are used in the MPLS network (for example, a TE tunnel between P routers or between P and PE routers, with LDP on the tunnel), the label stack is 3 (TE label, LDP label, VC label).
- If you use MPLS Fast Reroute in the MPLS network, you add a label to the stack. The maximum MPLS label stack in this case is 4 (FRR label, TE label, LDP label, VC label).

- If AToM is used by the customer carrier in the MPLS-VPN Carrier Supporting Carrier environment, you add a label to the stack. The maximum MPLS label stack in the provider carrier network is 5 (FRR label, TE label, LDP label, VPN label, VC label).
- If an AToM tunnel spans different service providers that exchange MPLS labels using IPv4 BGP (RFC 3107), you add a label to the stack. The maximum MPLS label stack is 5 (FRR label, TE label, BGP label, LDP label, VC label).

Other circumstances can increase the MPLS label stack size. Therefore, analyze the complete data path between the AToM tunnel endpoints and determine the maximum MPLS label stack size for your network. Then multiply the label stack size by the size of the MPLS label.

Estimating Packet Size—Example

Example 17-28 shows how to estimate the size of packets. The example uses the following assumptions:

- The edge MTU is 1500 bytes.
- The transport type is Ethernet VLAN, which designates 18 bytes for the transport header.
- The AToM header is 4 bytes, because the control word is always used.
- The MPLS label stack size is 2, because LDP is used. The MPLS label size is 4 bytes.

Example 17-28 Estimating the MTU for Packets

```
Core MTU >= (Edge MTU + Transport header + AToM header + (MPLS label stack size* MPLS
label size))
1500     + 18                    + 0      + (2                * 4          ) = 1526
```

You must configure the P and PE routers in the core to accept packets of 1526 bytes. See the following section for setting the MTU size on the P and PE routers.

Changing the MTU Size on P and PE Routers

After you determine the MTU size to set on your P and PE routers, you can issue the **mtu** command on the routers to set the MTU size. The following example specifies an MTU size of 1526 bytes.

```
Router(config-if)# mtu 1526
```

Setting Experimental Bits with AToM

MPLS AToM uses the three experimental (EXP) bits in a label to determine the queue of packets.The EXP bits are set to 0 (zero) by default. Table 17-7 summarizes the commands you can use to override the default values.

Table 17-7 Commands Supported to Change EXP Bits

Transport Type	Supported Commands
ATM AAL5 SDU support over MPLS Ethernet over MPLS: Port mode Frame Relay over MPLS: DLCI-to-DLCI connections Port-to-port connections HDLC over MPLS PPP over MPLS	**set mpls experimental** **match any**
Ethernet over MPLS: VLAN mode	**set mpls experimental** **match cos**

Set the experimental bits in both the VC label and the LSP tunnel label. You set the experimental bits in the VC label, because the LSP tunnel label might be removed at the penultimate router.

To set the experimental bits, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-name`	Specifies the user-defined name of the traffic class and enters class map configuration mode.

Step 2	For all transport types except Ethernet over MPLS in VLAN mode:
	`Router(config-cmap)# match any`
	
	For Ethernet over MPLS in VLAN mode only:
	`Router(config-cmap)# match cos cos-value`
		Specifies that all packets are matched.
		
		cos-value is from 0 to 7; up to four CoS values can be specified in one **match cos** statement.

Step 3	`Router(config-cmap)# policy-map policy-name`	Specifies the name of the traffic policy to configure and enters policy map configuration mode.
Step 4	`Router(config-pmap)# class class-name`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy. Enters policy-map configuration mode.
Step 5	`Router(config-pmap-c)# set mpls experimental value`	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 6	`Router(config-pmap-c)# exit`	Exits policy map configuration mode.
Step 7	`Router(config-pmap)# exit`	Exits policy map mode.
Step 8	`Router(config)# interface slot/port`	Specifies the interface and enters interface configuration mode.
Step 9	`Router(config-if)# service-policy input policy-name`	Attaches a traffic policy to an interface.

Displaying the Traffic Policy Assigned to an Interface

To display the traffic policy attached to an interface, use the **show policy-map interface** command.

Example 17-29 uses the **set mpls experimental** command with the **match any** command under a default class. This means that every packet tunneled onto a particular AToM VC carries the same MPLS experimental bit value.

Example 17-29 Setting EXP Bits Using the match any Command

```
class-map match-any default-class
  match any
policy-map atm-default-policy
  class default-class
    set mpls experimental 3
!
!
interface atm4/0
service-policy input atm-default-policy
```

Example 17-30 uses the **set mpls experimental** command with the **match cos** command. This allows packets tunneled onto a particular AToM VC to carry different MPLS experimental bit values. The **match cos** command is only configurable on Ethernet VLAN subinterfaces.

Example 17-30 Setting EXP Bits Using the match cos Command

```
class-map match-any match_cos_low
  match cos 0 1 2 3
class-map match-any match_cos_high
  match cos 4 5 6 7
policy-map ether-clp-policy
  class match_cos_low
    set mpls experimental 1
  class match_cos_high
    set mpls experimental 5
!
!
interface Gi0/0.1
service-policy input ether-clp-policy
```

Configuring QoS Features

For information about configuring QoS features on the Cisco 10000 series router, see the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Table 17-8 and Table 17-9 describe the policy map actions supported on various interfaces. The tables indicate the following:

- No—You cannot perform this policy map action or match criteria.
- Yes—You can perform this policy map action or match criteria.
- N/A (Not Applicable)—You can apply the policy map action or match criteria, but it does not have any effect on packets.

Table 17-8 Input (Imposition Router) Policy Map Actions

	Interface			
Policy Map Actions	**ATM**	**Ethernet**	**Frame Relay**	**HDLC and PPP**
bandwidth	no	no	no	no
queue-limit	no	no	no	no
priority	no	no	no	no
shape	no	no	no	no
random-detect	no	no	no	no
set ip prec/dscp	N/A	N/A	N/A	N/A
set qos-group	yes	yes	yes	yes
set discard class	yes	yes	yes	yes
set atm-clp	N/A	N/A	N/A	N/A
set fr-de	N/A	N/A	N/A	N/A
set cos	N/A	N/A	N/A	N/A
police	yes	yes	yes	yes
set mpls-exp topmost	N/A	N/A	N/A	N/A
set mpls-exp imposition	yes	yes	yes	yes

Table 17-9 Output (Disposition Router) Policy Map Actions

	Interface			
Policy Map Actions	**ATM**	**Ethernet**	**Frame Relay**	**HDLC and PPP**
bandwidth	yes	yes	yes	yes
queue-limit	yes	yes	yes	yes
priority	yes	yes	yes	yes
shape	yes	yes	yes	yes
random-detect	yes (discard class only)	yes (discard class only)	yes (discard class only)	yes (discard class only)
set ip prec/dscp	no	no	no	no
set qos-group	N/A	N/A	N/A	N/A
set discard class	no	no	no	no
set atm-clp	yes	no	no	no
set fr-de	no	no	no	no
set cos	no	yes	no	no
police	yes	yes	yes	yes
set mpls-exp topmost	no	no	no	no
set mpls-exp imposition	N/A	N/A	N/A	N/A

Table 17-10 and Table 17-11 describe support for class map match criteria on various interfaces. Table 17-10 describes match criteria support for inbound traffic and Table 17-11 describes support for outbound traffic.

Table 17-10 Input (Imposition Router) Class Map Match Criteria

	Interface			
Match Criteria	**ATM**	**Ethernet**	**Frame Relay**	**HDLC and PPP**
DSCP	no	no	no	no
IP precedence	no	no	no	no
MPLS EXP	no	no	no	no
IEE 802.1P bits	no	yes	no	no
Access-list	no	no	no	no
QoS group	N/A	N/A	N/A	N/A
Discard class	N/A	N/A	N/A	N/A
Input interface	yes	yes	yes	yes
Protocol	no	no	no	no
RTP	no	no	no	no
atm-clp	no	no	no	no
MAC address	no	no	no	no
Frame Relay DLCI	no	no	no	no
VLAN ID	no	no	no	no
Packet length	no	no	no	no
DE bit (Frame Relay)	no	no	no	no

Table 17-11 Output (Disposition Router) Class Map Match Criteria

	Interface			
Match Criteria	**ATM**	**Ethernet**	**Frame Relay**	**HDLC and PPP**
DSCP	no	no	no	no
IP precedence	no	no	no	no
MPLS EXP	N/A	N/A	N/A	N/A
IEEE 802.1P bits	N/A	N/A	N/A	N/A
Access-list	no	no	no	no
QoS group	yes	yes	yes	yes
Discard class	yes	yes	yes	yes
Input interface	yes	yes	yes	yes
Protocol	no	no	no	no
RTP	no	no	no	no
atm-clp	N/A	N/A	N/A	N/A
MAC address	no	no	no	no

Table 17-11 Output (Disposition Router) Class Map Match Criteria (continued)

	Interface			
Match Criteria	**ATM**	**Ethernet**	**Frame Relay**	**HDLC and PPP**
Frame Relay DLCI	no	no	no	no
VLAN ID	no	no	no	no
Packet Length	no	no	no	no
DE bit (Frame Relay)	N/A	N/A	N/A	N/A

Monitoring and Maintaining L2VPN

To monitor and maintain the configuration of L2VPN features, use the following commands in privileged EXEC mode. Note that with the exception of the **show mpls l2transport** command, these commands can produce output that is meant to be used by Cisco Systems technical support personnel only.

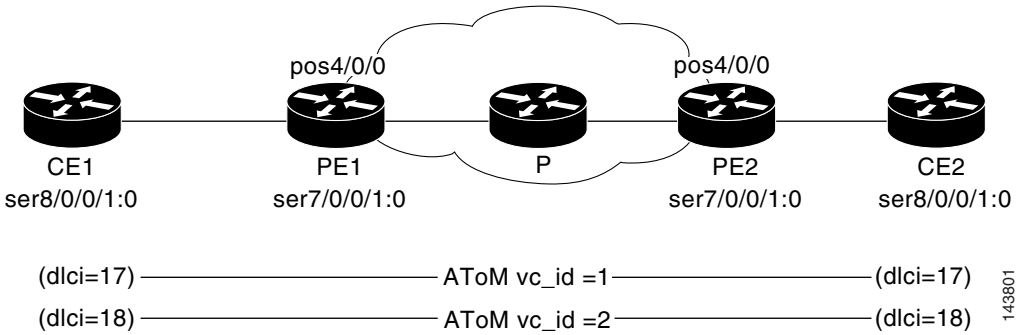
Command	Displays
show mpls l2transport	Information about AToM VCs that have been enabled to route Layer 2 packets on a router, including platform-independent AToM status and the AToM capabilities of a particular interface.
show pxf cpu atom	PXF-specific forwarding AToM and LS information for an interface or VCCI (column 1 forwarding information).
show mpls l2transport vc	Information about AToM virtual circuits (VCs) that have been enabled to route Layer 2 packets on a router.
show pxf cpu mpls label	PXF-specific forwarding information for a label. The output has been extended to indicate AToM disposition labels, specifically, the transport type associated with the label and the set of output features associated with the label, such as control word and sequencing.
show pxf cpu subblocks	Status and PXF-related parameters for the interface and has been extended to display column 0 of AToM status.
show ssm	Platform-specific information about active segments.
debug pxf atom ac	AToM information related to attachment circuit events.
debug pxf atom mpls	AToM information related to MPLS Forwarding Information (MFI)-driven events.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco Systems technical support personnel. Moreover, it is best to use **debug** commands during periods of low network traffic and few users. This decreases the likelihood that increased **debug** command processing overhead will affect system use.

Configuration Example—Frame Relay over MPLS

Example 17-31 shows the configuration of Frame Relay over MPLS on two provider edge routers, PE1 and PE2, and on two customer edge routers, CE1 and CE2. The topology for the example is shown in Figure 17-7.

Figure 17-7 Frame Relay over MPLS Example Topology



For the AToM VCs to come up, MPLS/LDP and a routing protocol need to be run in the core network (PE1---P----PE2). PE1 and PE2 show that they are enabled with the OSPF routing protocol and MPLS/LDP.

Example 17-31 Frame Relay over MPLS Configuration

CE1 Configuration for Frame Relay

```
================================
interface Serial8/0/0.1/1:0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay lmi-type q933a
frame-relay intf-type dce
interface Serial8/0/0.1/1:0.1 point-to-point
ip address 192.1.1.1 255.255.255.0
frame-relay interface-dlci 17
!
interface Serial8/0/0.1/1:0.2 point-to-point
ip address 192.1.2.1 255.255.255.0
frame-relay interface-dlci 18
```

PE1 Configuration for LDP and AToM VC

```
================================
!Enabling LDP
mpls ldp graceful-restart timers neighbor-liveness 300
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!Define Loopback address for LDP protocol
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!Enable MPLS/LDP on the core interface
interface POS4/0/0
ip address 50.0.0.1 255.0.0.0
mpls label protocol ldp
```

```
mpls ip
crc 32
clock source internal
!
!Enabling OSPF protocol
router ospf 100
log-adjacency-changes
network 1.0.0.0 0.255.255.255 area 100
network 50.0.0.0 0.255.255.255 area 100
!Define pseudowire-class
pseudowire-class pw_atom1
encapsulation mpls
!FR configuration with two subinterfaces
interface Serial8/0/0.1/1:0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay lmi-type q933a
!
interface Serial8/0/0.1/1:0.1 point-to-point
!
interface Serial8/0/0.1/1:0.2 point-to-point
!
!Two AToM VC configuration with vc ids 1 & 2, 2.2.2.2 is LB addr of PE2
connect atom1 Serial8/0/0.1/1:0 17 l2transport
xconnect 2.2.2.2 1 pw-class pw_atom1
!
!
connect atom2 Serial8/0/0.1/1:0 18 l2transport
xconnect 2.2.2.2 2 pw-class pw_atom1
```

PE2 Configuration

```
================================
!Enabling LDP
mpls ldp graceful-restart timers neighbor-liveness 300
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!Define Loopback address for LDP protocol
interface Loopback0
ip address 2.2.2.2 255.255.255.255
!Enable MPLS/LDP on the core interface
interface POS4/0/0
ip address 60.0.0.2 255.0.0.0
mpls label protocol ldp
mpls ip
crc 32
clock source internal
!
!Enabling OSPF protocol
router ospf 100
log-adjacency-changes
network 2.0.0.0 0.255.255.255 area 100
network 60.0.0.0 0.255.255.255 area 100
!Define pseudowire-class
pseudowire-class pw_atom1
encapsulation mpls
!FR configuration with two subinterfaces
interface Serial8/0/0.1/1:0
no ip address
encapsulation frame-relay
no fair-queue
```

```
frame-relay lmi-type q933a
interface Serial8/0/0.1/1:0.1 point-to-point
interface Serial8/0/0.1/1:0.2 point-to-point
!Two AToM VC configuration with vc ids 1 & 2
connect atom1 Serial8/0/0.1/1:0 17 l2transport
xconnect 1.1.1.1 1 pw-class pw_atom1
!
!
connect atom2 Serial8/0/0.1/1:0 18 l2transport
xconnect 1.1.1.1 2 pw-class pw_atom1
```

CE2 Configuration

```
================================
interface Serial8/0/0.1/1:0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay lmi-type q933a
frame-relay intf-type dce
!
interface Serial8/0/0.1/1:0.1 point-to-point
ip address 192.1.1.2 255.255.255.0
frame-relay interface-dlci 17
!
interface Serial8/0/0.1/1:0.2 point-to-point
ip address 192.1.2.2 255.255.255.0
frame-relay interface-dlci 18
```

Verifying PE1 Configuration

The PE1 router shows two AToM VCs are up.

```
================================
router# show mpls l2tran vc
Local intf Local circuit Dest address VC ID Status
------------ ------------------------- --------------- ---------- ----------
Se8/0/0.1/1:0 FR DLCI 17 2.2.2.2 1 UP
Se8/0/0.1/1:0 FR DLCI 18 2.2.2.2 2 UP

router# show mpls l2tran vc 1 det
Local interface: Se8/0/0.1/1:0 up, line protocol up, FR DLCI 17 up
Destination address: 2.2.2.2, VC ID: 1, VC status: up
Output interface: PO4/0/0, imposed label stack {93 19}
Preferred path: not configured
Default path: active
Next hop: point2point
Create time: 00:00:49, last status change time: 00:00:06
Signaling protocol: LDP, peer 2.2.2.2:0 up
MPLS VC labels: local 19, remote 93
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description:
Sequencing: receive disabled, send disabled
VC statistics:
packet totals: receive 0, send 0
byte totals: receive 0, send 0
packet drops: receive 0, seq error 0, send 0

router# sh mpls l2tran vc 2 det
Local interface: Se8/0/0.1/1:0 up, line protocol up, FR DLCI 18 up
Destination address: 2.2.2.2, VC ID: 2, VC status: up
Output interface: PO4/0/0, imposed label stack {98 19}
Preferred path: not configured
```

```
Default path: active
Next hop: point2point
Create time: 00:00:53, last status change time: 00:00:10
Signaling protocol: LDP, peer 2.2.2.2:0 up
MPLS VC labels: local 22, remote 98
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description:
Sequencing: receive disabled, send disabled
VC statistics:
packet totals: receive 0, send 0
byte totals: receive 0, send 0
packet drops: receive 0, seq error 0, send 0
```

Any Transport over MPLS—Tunnel Selection

Tunnel Selection allows you to specify the path that Any Transport over MPLS (AToM) traffic uses. You can specify either a MPLS traffic engineering tunnel or a destination IP address. If the specified path is unreachable, you can specify that the virtual circuits (VCs) should use the default path, which is the path that MPLS Label Distribution Protocol (LDP) uses for signaling.

Note By default, the **preferred-path** sub-command has a fallback pseudowire. If the preferred pseudowire goes down, the MPLS/LDP module switch the circuit temporarily to another pseudowire. When the preferred pseudowire is up again, the circuit is switched back to the preferred pseudowire. The **preferred-path** subcommand also has an **disable-fallback** option, so that no random pseudowire is chosen if the preferred path goes down. The circuit is down until the preferred path pseudowire comes back up. However, in the 12.2(33) SB release, by default, the **preferred-path** sub-command has the **disable-fallback** option. There is no fallback pseudowire in this release, even when the option is stated explicitly.

See the Any Transport over MPLS: Tunnel Selection document for the following information:

- Prerequisites for Any Transport over MPLS: Tunnel Selection
- Restrictions for Any Transport over MPLS: Tunnel Selection
- Configuring Any Transport over MPLS: Tunnel Selection
- The **debug mpls l2transport vc** command for verifying Any Transport over MPLS: Tunnel Selection
- Verifying the Configuration—Example
- Troubleshooting Any Transport over MPLS: Tunnel Selection—Example

Configuration Example—Any Transport over MPLS: Tunnel Selection

The following example sets up two preferred paths for PE1. One preferred path specifies an MPLS traffic engineering tunnel. The other preferred path specifies an IP address of a loopback address on PE2. There is a static route configured on PE1 that uses a TE tunnel to reach the IP address on PE2.

Router PE1

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback0
```

```
pseudowire-class pw1
encapsulation mpls
preferred-path interface Tunnel1 disable-fallback
!
pseudowire-class pw2
encapsulation mpls
preferred-path peer 10.18.18.18
!
interface Loopback0
ip address 10.2.2.2 255.255.255.255
no ip directed-broadcast
no ip mroute-cache
!
interface Tunnel1
ip unnumbered Loopback0
no ip directed-broadcast
tunnel destination 10.16.16.16
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 1500
tunnel mpls traffic-eng path-option 1 explicit name path-tu1
!
interface Tunnel2
ip unnumbered Loopback0
no ip directed-broadcast
tunnel destination 10.16.16.16
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 1500
tunnel mpls traffic-eng path-option 1 dynamic
!
interface gigabitethernet0/0/0
no ip address
no ip directed-broadcast
no negotiation auto
!
interface gigabitethernet0/0/0.1
encapsulation dot1Q 222
no ip directed-broadcast
xconnect 10.16.16.16 101 pw-class pw1
!
interface ATM1/0/0
no ip address
no ip directed-broadcast
no atm enable-ilmi-trap
no atm ilmi-keepalive
pvc 0/50 l2transport
encapsulation aal5
xconnect 10.16.16.16 150 pw-class pw2
!
interface gigabitEthernet2/0/1
ip address 10.0.0.1 255.255.255.0
no ip directed-broadcast
tag-switching ip
mpls traffic-eng tunnels
ip rsvp bandwidth 15000 15000
!
router ospf 1
log-adjacency-changes
network 10.0.0.0 0.0.0.255 area 0
network 10.2.2.2 0.0.0.0 area 0
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
!
```

```
ip route 10.18.18.18 255.255.255.255 Tunnel2
!
ip explicit-path name path-tu1 enable
next-address 10.0.0.1
index 3 next-address 10.0.0.1
```

Router PE2

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback0
interface Loopback0
ip address 10.16.16.16 255.255.255.255
no ip directed-broadcast
no ip mroute-cache
!
interface Loopback2
ip address 10.18.18.18 255.255.255.255
no ip directed-broadcast
!
interface gigabitEthernet3/1
ip address 10.0.0.2 255.255.255.0
no ip directed-broadcast
mpls traffic-eng tunnels
mpls ip
no cdp enable
ip rsvp bandwidth 15000 15000
!
interface gigabitEthernet3/3
no ip address
no ip directed-broadcast
no cdp enable
!
interface gigabitEthernet3/3.1
encapsulation dot1Q 222
no ip directed-broadcast
no cdp enable
mpls l2transport route 10.2.2.2 101
!
interface ATM5/0
no ip address
no ip directed-broadcast
no atm enable-ilmi-trap
no atm ilmi-keepalive
pvc 0/50 l2transport
encapsulation aal5
xconnect 10.2.2.2 150 encapsulation mpls
!
router ospf 1
log-adjacency-changes
network 10.0.0.0 0.0.0.255 area 0
network 10.16.16.16 0.0.0.0 area 0
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
```

CHAPTER 18

Configuring L2VPN Interworking

Interworking is the transforming function required to interconnect two heterogeneous alternating currents (ACs). Several types of interworking functions exist. The function that is used depends on the type of ACs being used, the type of data being carried, and the level of functionality required. The two main interworking functions supported in Cisco IOS software are:

- Bridged Interworking—Used when only Layer 2 (L2) packets are considered without regard to Layer 3 contents. No routing participation by the Internet Service Provider (ISP) exists. In particular, the software supports the use of the Ethernet (port) over MPLS pseudowire for bridged interworking. For this reason, this type of interworking function is also called Ethernet Interworking.
- Routed Interworking—Used to carry Layer 3 packets. A different routed interworking function exists for each protocol type. The most common routed interworking function supports Internet Protocol (IP). Therefore, this type of interworking function is also called IP Interworking, and a new type of pseudowire, IP over MPLS, is used.

Note Cisco 10000 series of routers only support Ethernet interworking.

Layer 2 transport over MPLS and IP already exists for like-to-like attachment circuits, such as Ethernet-to-Ethernet or PPP-to-PPP. L2VPN Interworking builds on this functionality by allowing disparate attachment circuits to be connected. An interworking function facilitates the translation between different Layer 2 encapsulations. Several interworking modes exist, but the Cisco 10000 series router supports only bridged interworking, also known as Ethernet interworking.

To specify a mode, issue the **interworking** {**ethernet** | **ip**} command in pseudowire-class configuration mode. The **interworking** command causes the attachment circuits to be terminated locally. The **ethernet** keyword causes Ethernet frames to be extracted from the attachment circuit and sent over the pseudowire. Ethernet end-to-end transmission is assumed. Attachment circuit frames that are not Ethernet are dropped. In the case of VLAN, the VLAN tag is removed, leaving an untagged Ethernet frame.

This chapter describes the following L2VPN like-to-like and interworking features:

Ethernet to VLAN—Bridged Interworking

In Ethernet Interworking, also called as bridged interworking, Ethernet frames are bridged across the pseudowire. The customer edge (CE) routers could be natively bridging Ethernet or could be routing using a bridged encapsulation model, such as Bridge Virtual Interface (BVI) or RBE. The PE routers operate in Ethernet like-to-like mode.

The Ethernet to VLAN (Bridged) feature is described in the following topics:

Configuring L2VPN Interworking

Enabling L2VPN Interworking requires that you add the **interworking** command to the list of commands that comprise the pseudowire.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**
5. **interworking ethernet**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. if prompted, enter your password.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class class1`	Establishes a pseudowire class with a name that you specify. Enters pseudowire class configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 5	**interworking ethernet** **Example:** `Router(config-pw)# interworking ethernet`	Specifies the type of pseudowire and the type of traffic that can flow across it.

Verifying the Configuration

You can verify the AToM configuration by using the **show mpls l2transport vc detail** command. In the following example, the interworking type appears in bold.

PE1	PE2

`Router# show mpls l2transport vc detail`
 	
`Local interface: Fa1/1/0 up, line protocol up, Ethernet up`
`  Destination address: 10.9.9.9, VC ID: 123, VC status: up`
`    Preferred path: not configured`
`    Default path: active`
`    Tunnel label: 17, next hop 10.1.1.3`
`    Output interface: Fa4/0/0, imposed label stack {17 20}`
`  Create time: 01:43:50, last status change time: 01:43:33`
`  Signaling protocol: LDP, peer 10.9.9.9:0 up`
`    MPLS VC labels: local 16, remote 20`
`    Group ID: local 0, remote 0`
`    MTU: local 1500, remote 1500`
`    Remote interface description:`
`  Sequencing: receive disabled, send disabled`
`  VC statistics:`
`    packet totals: receive 15, send 4184`
`    byte totals:   receive 1830, send 309248`
`    packet drops:  receive 0, send 0`
	`Router# show mpls l2transport vc detail`
	
	`Local interface: Fa2/0.3 up, line protocol up, Eth VLAN 10 up`
	`  MPLS VC type is Ethernet, interworking type is Ethernet`
	`  Destination address: 10.8.8.8, VC ID: 123, VC status: up`
	`    Preferred path: not configured`
	`    Default path: active`
	`    Tunnel label: 16, next hop 10.1.1.3`
	`    Output interface: Fa6/0, imposed label stack {16 16}`
	`  Create time: 00:00:26, last status change time: 00:00:06`
	`  Signaling protocol: LDP, peer 10.8.8.8:0 up`
	`    MPLS VC labels: local 20, remote 16`
	`    Group ID: local 0, remote 0`
	`    MTU: local 1500, remote 1500`
	`    Remote interface description:`
	`  Sequencing: receive disabled, send disabled`
	`  VC statistics:`
	`    packet totals: receive 5, send 0`
	`    byte totals:   receive 340, send 0`
	`    packet drops:  receive 0, send 0`

Configuration Examples of Ethernet to VLAN—Bridged

This section contains examples of Ethernet to VLAN for both local switching (LS) and AToM:

- Ethernet to VLAN over LS—Bridged: Example
- Ethernet to VLAN over AToM—Bridged: Example

Ethernet to VLAN over LS—Bridged: Example

PE
config t interface atm 2/0/0 pvc 0/200 l2transport encapsulation aal5snap interface gigabitethernet 5/1/0 no ip address connect ETH-VLAN gigabitethernet 5/0/0 gigabitethernet 5/1/0.3 interworking ethernet

Ethernet to VLAN over AToM—Bridged: Example

PE1	PE2
ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom encapsulation mpls ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet1/0 xconnect 10.9.9.9 123 pw-class atom	ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom-eth-iw encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0.1 encapsulation dot1q 100 xconnect 10.9.9.9 123 pw-class atom-eth-iw

Ethernet/VLAN to ATM AAL5 Interworking

The Ethernet/VLAN to ATM AAL5 Interworking feature is described in the following topics:

Prerequisites of Ethernet/VLAN to ATM AAL5 Interworking

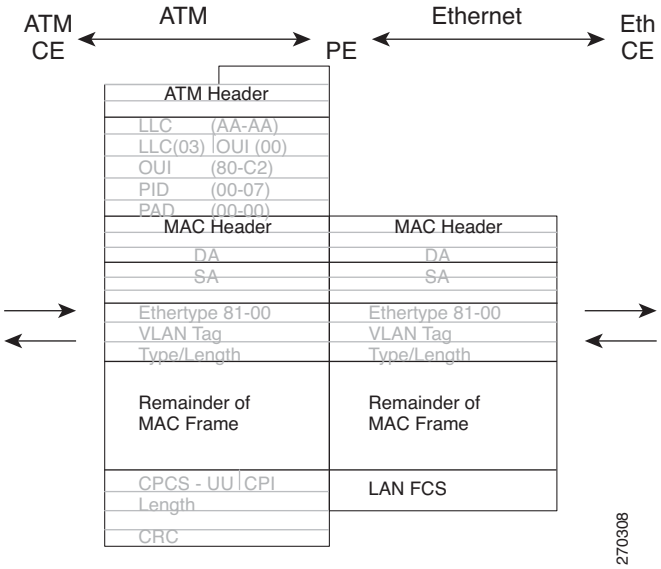
Before you configure Ethernet/VLAN to ATM AAL5 Interworking on a network, you must enable Cisco Express Forwarding.

Restrictions of Ethernet/VLAN to ATM AAL5 Interworking

In Cisco IOS Release 12.2(33)SB, the Ethernet/VLAN to ATM AAL5 local switching has the following restrictions:

- The following translations are only supported and other translations are dropped:
  - Ethernet without LAN FCS (AAAA030080C200070000)
  - Spanning tree (AAAA030080C2000E)
- ATM encapsulation types supported for bridged interworking: aal5snap.
- The existing QoS functionality for ATM is supported, including setting the ATM CLP bit.
- Only ATM AAL5 virtual circuit (VC) mode is supported. ATM VP and port mode are not supported.
- The non-AAL5 traffic is punted, for example, OAM cells. The end-to-end F5 loopback cells are looped back onto the PE router.
- If the Ethernet frame arriving from Ethernet CE includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame and it is forwarded to the ATM CE, as shown in Figure 18-1.

Figure 18-1 Protocol Stack for ATM AAL5 to Ethernet Local Switching Bridged Interworking—With VLAN Header

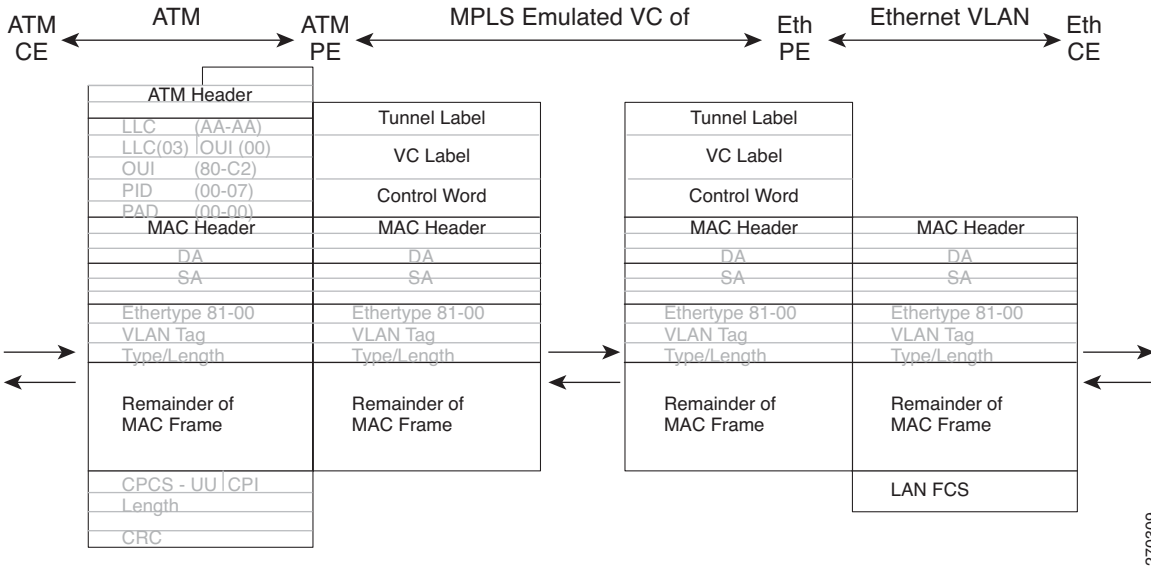


In Cisco IOS Release 12.2(33)SB, the Ethernet/VLAN to ATM AAL5 AToM has the following restrictions:

- The following translations are only supported and other translations are dropped:
  - Ethernet without LAN FCS (AAAA030080C200070000)
  - Spanning tree (AAAA030080C2000E)

- ATM encapsulation types supported for bridged interworking: aal5snap.
- The existing QoS functionality for ATM is supported, including setting the ATM CLP bit.
- Only ATM AAL5 VC mode is supported. ATM VP and port mode are not supported.
- SVCs are not supported.
- Individual AAL5 ATM cells are assembled into frames before being sent across the pseudowire.
- Non-AAL5 traffic, (such as OAM cells) is punted to be processed at RP level. A VC that has been configured with OAM cell emulation on the ATM PE router (using the **oam-ac emulation-enable** CLI command) can send end-to-end F5 loopback cells at configured intervals toward the CE router.
- When the pseudowire is down, an F5 end-to-end segment AIS/RDI (Alarm indication signal/Remote defect indication) is sent from the PE router to the CE router.
- If the Ethernet frame arriving from Ethernet CE includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame across the pseudowire (Figure 18-2).

Figure 18-2 Protocol Stack for ATM to Ethernet AToM Bridged Interworking—With VLAN Header



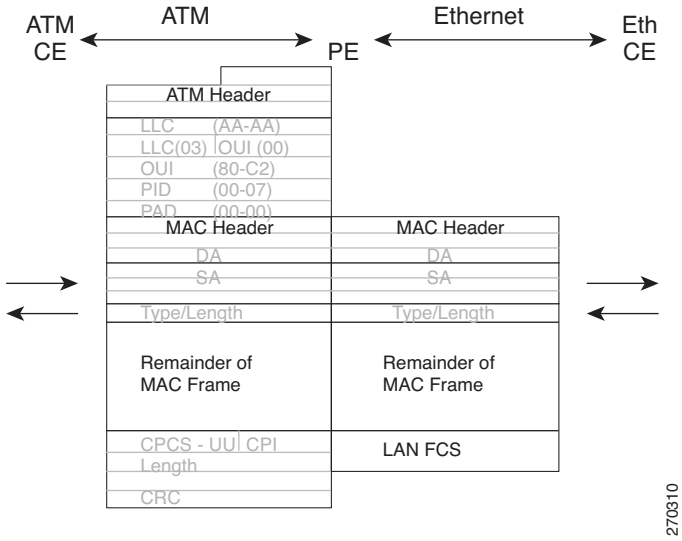
ATM AAL5 to Ethernet Local Switching—Bridged Interworking

This interworking type provides interoperability between Ethernet attachment VC and ATM attachment VC connected to the same PE router. For this interworking type, Bridged encapsulation is used, corresponding to the Bridged Interworking mechanism.

- In Ethernet to ATM direction, the PE router forwards the Layer 2 packet without any change to the egress interface, encapsulating the Layer 2 packet over AAL5 using Bridged encapsulation.
- In ATM to Ethernet direction, the ATM header and bridged encapsulation get discarded and the Layer 2 packet is sent out with Ethernet encapsulation.

Figure 18-3 shows the protocol stack for ATM to Ethernet local switching -bridged interworking. The ATM side has an encapsulation type as aal5snap.

Figure 18-3 Protocol Stack for ATM AAL5 to Ethernet Local Switching Bridged Interworking



ATM AAL5 to VLAN 802.1Q Local Switching—Bridged Interworking

This interworking type provides interoperability between ATM attachment VC and Ethernet VLAN attachment VC connected to the same PE router. As in the ATM to Ethernet case, Bridged encapsulation is used, corresponding to Bridged (Ethernet) Interworking mechanism.

In case of Ethernet VLAN attachment, the VLAN ID is a service delimiter, so the VLAN header is not included in the frame to and from the ATM CE.

- In the VLAN to ATM direction, the PE router discards the VLAN header from the Layer 2 packet. The PE router sends the frame to the ATM egress interface after encapsulating the L2 packet over AAL5 using Bridged encapsulation.
- In the ATM to VLAN direction, the ATM header and bridged encapsulation are discarded and the L2 packet is sent out with a VLAN header inserted following the destination/source MAC addresses.

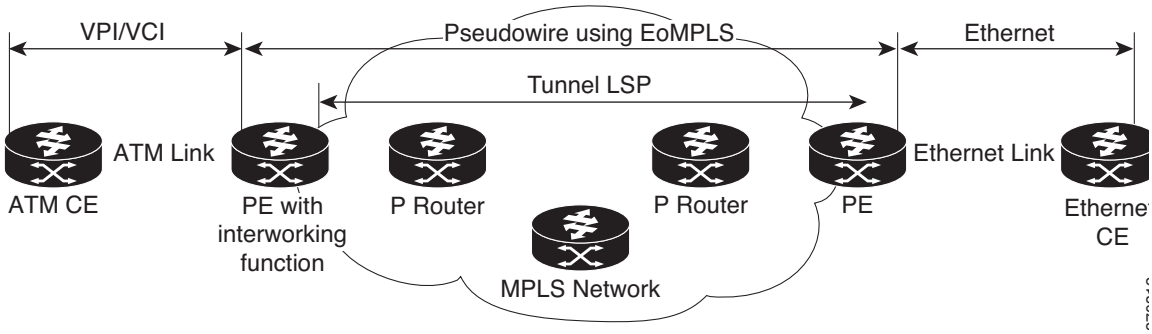
The protocol stack for ATM to VLAN local switching is shown in Figure 18-3. The ATM side has an encapsulation type of aal5snap.

ATM AAL5 to Ethernet Port AToM—Bridged Interworking

This interworking type provides interoperability between ATM attachment VC and Ethernet attachment VC connected to different PE routers. Bridged encapsulation is used, corresponding to the Bridged (Ethernet) Interworking mechanism.

The interworking function is performed at the PE connected to the ATM attachment VC based on Multiprotocol Encapsulation over ATM Adaptation Layer 5 (Figure 18-4).

Figure 18-4 Network Topology for ATM to Ethernet AToM Bridged Interworking



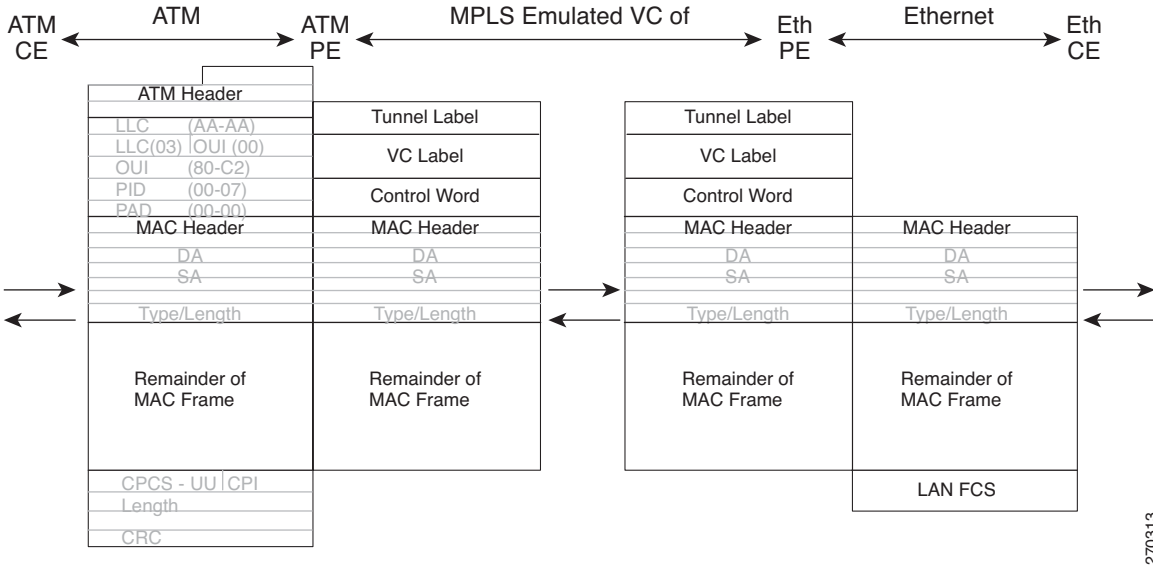
The advantage of this architecture is that the Ethernet PE (connected to the Ethernet segment) operates similarly to Ethernet like-to-like services.

On the PE with Interworking function, in the direction from the ATM segment to MPLS cloud, the bridged encapsulation (ATM/SNAP header) is discarded and the Ethernet frame is encapsulated with the labels required to go through the pseudowire using the VC type 5 (Ethernet) (Figure 18-5).

In the opposite direction, after the label disposition from the MPLS cloud, Ethernet frames are encapsulated over AAL5 using bridged encapsulation.

Figure 18-5 shows the protocol stack for ATM to Ethernet AToM Bridged Interworking. The ATM side has an encapsulation type of aal5snap.

Figure 18-5 Protocol Stack for ATM to Ethernet AToM Bridged Interworking—Without VLAN Header



ATM AAL5 to Ethernet VLAN 802.1Q AToM—Bridged Interworking

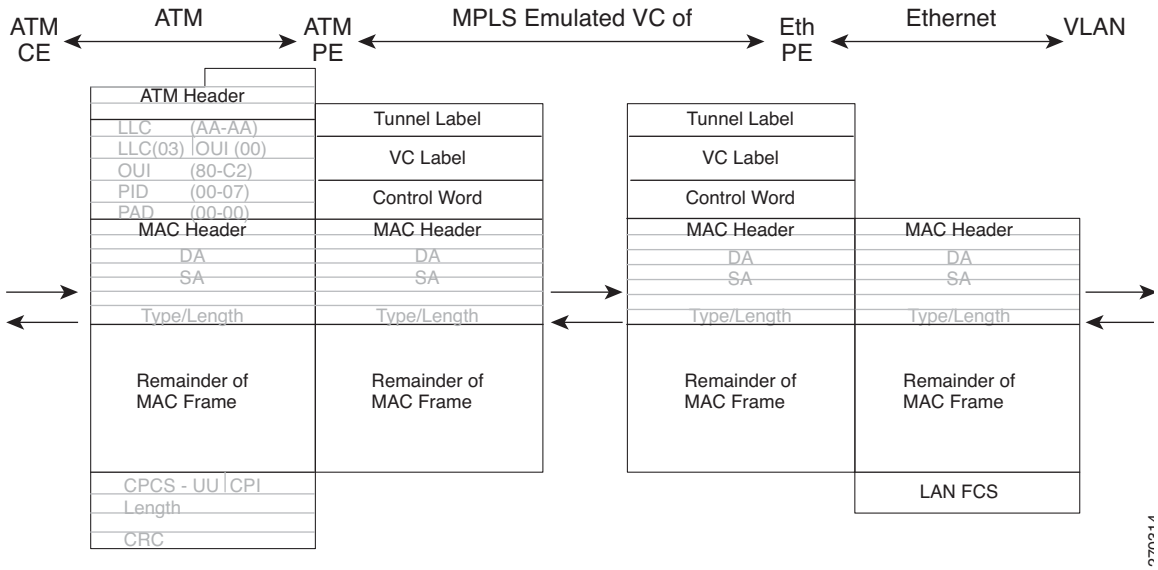
This interworking type provides interoperability between ATM attachment VC and Ethernet VLAN attachment VC connected to different PE routers. Bridged encapsulation is used, corresponding to the Bridged (Ethernet) Interworking mechanism.

The interworking function is performed in the same way as for the ATM to Ethernet Port case, implemented on the PE connected to the ATM attachment VC. The implementation is based on Multiprotocol Encapsulation over ATM Adaptation Layer 5 (see Figure 18-4).

For the PE connected to the Ethernet side, one major difference exists due the existence of the VLAN header in the incoming packet. The PE discards the VLAN header of the incoming frames from the VLAN CE, and the PE inserts a VLAN header into the Ethernet frames traveling from the MPLS cloud. The frames sent on the pseudowire (with VC type 5) are Ethernet frames without the VLAN header.

Encapsulation over ATM Adaptation Layer 5, as shown in Figure 18-6.

Figure 18-6 Protocol Stack for ATM to VLAN AToM Bridged Interworking



Configuration Tasks and Examples

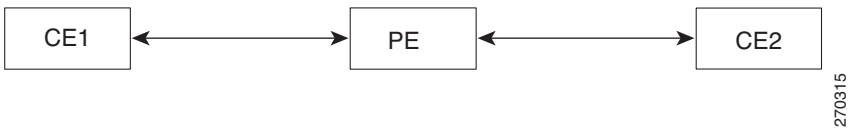
This section describes configuration tasks for and provides examples of two L2VPN technology solutions:

- Local Switching
- AToM

Local Switching

Figure 18-7 shows different LS configurations.

Figure 18-7 Local Switching Model for CLI Commands



Note that LS interworking on the Cisco 10000 router only supports the Bridged Interworking function, also known as Ethernet interworking function.

This section explains the following LS configurations and their examples:

ATM AAL5 to Ethernet Port

You can configure the ATM AAL5 to Ethernet Port feature on a PE router using the following steps:

1. **config t**
2. **interface atm** slot/subslot/port
3. **pvc** vpi/vci **l2transport**
4. **encapsulation aal5snap**
5. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port
6. **no ip address**
7. **connect** connection-name [**fastethernet** | **gigabitethernet**] slot/subslot/port **atm** slot/subslot/port vpi/vci **interworking ethernet**

Note The order of the interfaces in the **connect** command is not important.

The following example shows how you can configure the ATM AAL5 to Ethernet Port feature on a PE router:

```
config t
interface atm 2/0/0
    pvc 0/200 l2transport
    encapsulation aal5snap
interface gigabitethernet 5/1/0
    no ip address
connect atm-enet gigabitethernet 5/1/0 atm 2/0/0 0/200 interworking ethernet
```

ATM AAL5 to Ethernet VLAN 802.1Q

You can configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE router using the following steps:

1. **config t**
2. **interface atm** slot/subslot/port
3. **pvc** vpi/vci **l2transport**
4. **encapsulation aal5snap**
5. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface
6. **encapsulation dot1q** VLAN-ID
7. **connect** connection-name [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface **atm** slot/subslot/port vpi/vci **interworking ethernet**

Note The order of the interfaces in the **connect** command is not important.

The following example shows how to configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE router:

```
config t
```

```
interface atm 2/0/0
    pvc 0/200 l2transport
    encapsulation aal5snap
interface gigabitethernet 5/1/0.3
    encapsulation dot1q 2
connect atm-vlan gigabitethernet 5/1/0.3 atm 2/0/0 0/200 interworking ethernet
```

AToM

Figure 18-8 illustrates different AToM configurations.

Figure 18-8 AToM Model for CLI Commands

Note that AToM interworking for Cisco 10000 routers only supports the bridged interworking function, also known as Ethernet interworking function.

This section explains the following AToM configurations and their examples:

- ATM AAL5 to Ethernet Port, page 18-11
- Configuring ATM AAL5 to Ethernet VLAN 802.1Q, page 18-12

ATM AAL5 to Ethernet Port

You can configure the ATM AAL5 to Ethernet Port feature on a PE1 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class** name
6. **encapsulation mpls**
7. **interworking ethernet**
8. **interface atm** slot/subslot/port
9. **pvc** vpi/vci **l2transport**
10. encapsulation aal5snap
11. **xconnect** remote-ip-address vc-id **pw-class** name

You can configure the ATM AAL5 to Ethernet Port feature on a PE2 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class** name
6. **encapsulation mpls**

7. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port
8. **xconnect** remote-ip-address vc-id **pw-class** name

Note The PE2 configuration does not include the **interworking ethernet** command because it is treated as like-to-like, and because the attachment circuit is already Ethernet port.

The following example shows how to configure the ATM AAL5 to Ethernet Port feature on a PE1 router:

```
config t
mpls label protocol ldp
interface Loopback100
 ip address 10.0.0.100 255.255.255.255
pseudowire-class atm-eth
 encapsulation mpls
 interworking ethernet
interface atm 2/0/0
 pvc 0/200 l2transport
 encapsulation aal5snap
 xconnect 10.0.0.200 140 pw-class atm-eth
```

The following example shows how to configure the ATM AAL5 to Ethernet Port feature on a PE2 router:

```
config t
mpls label protocol ldp
interface Loopback200
 ip address 10.0.0.200 255.255.255.255
pseudowire-class atm-eth
 encapsulation mpls
interface gigabitethernet 5/1/0
 xconnect 10.0.0.100 140 pw-class atm-eth
```

Configuring ATM AAL5 to Ethernet VLAN 802.1Q

You can configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE1 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class** name
6. **encapsulation mpls**
7. **interworking ethernet**
8. **interface atm** slot/subslot/port
9. **pvc** vpi/vci **l2transport**
10. **encapsulation aal5snap**
11. **xconnect** remote-ip-address vc-id **pw-class** name

You can configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE2 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**

3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class** name
6. **encapsulation mpls**
7. **interworking ethernet**
8. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface
9. **encapsulation dot1q** VLAN-ID
10. **xconnect** remote-ip-address vci **pw-class** name

Note In the case of ATM AAl5 to VLAN, the PE2 configuration does include the **interworking ethernet** command.

The following example shows how to configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE1 router:

```
config t
mpls label protocol ldp
interface Loopback100
 ip address 10.0.0.100 255.255.255.255
pseudowire-class atm-vlan
 encapsulation mpls
 interworking ethernet
interface atm 2/0/0
 pvc 0/200 l2transport
 encapsulation aal5snap
 xconnect 10.0.0.200 140 pw-class atm-vlan
```

The following example shows how to configure the ATM AAL5 to Ethernet VLAN 802.1Q feature on a PE2 router:

```
config t
mpls label protocol ldp
interface Loopback200
 ip address 10.0.0.200 255.255.255.255
pseudowire-class atm-vlan
 encapsulation mpls
 interworking ethernet
interface gigabitethernet 5/1/0.3
 encapsulation dot1q 1525
 xconnect 10.0.0.100 140 pw-class atm-vlan
```

Note To verify the L2VPN interworking status and check the statistics, refer to the "Verifying L2VPN Interworking" section on page 18-22.

Ethernet/VLAN to Frame Relay Interworking

The Ethernet VLAN to Frame Relay (FR) Interworking feature is described in the following topics:

Prerequisites of Ethernet/VLAN to Frame Relay Interworking

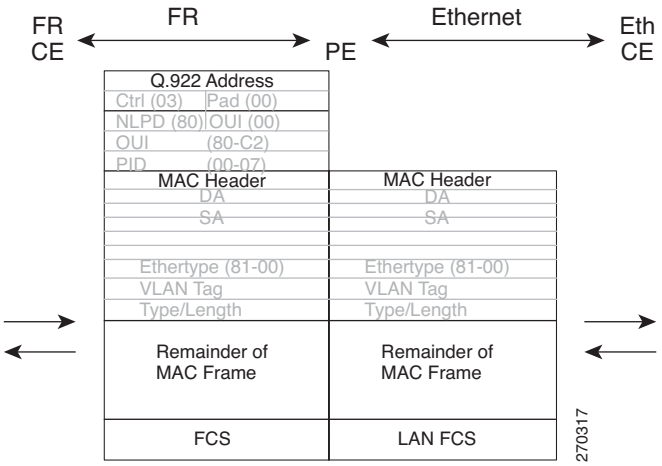
Before you configure Ethernet/VLAN to Frame Relay Interworking on a network, you must enable Cisco Express Forwarding.

Restrictions for Ethernet/VLAN to Frame Relay Interworking

In Cisco IOS Release 12.2(33)SB, the Ethernet/VLAN to Frame Relay LS has the following restrictions:

- The following translations are only supported and other translations are dropped:
  - Ethernet without LAN FCS (0300800080C20007 or 6558)
  - Spanning tree (0300800080C2000E or 4242)
- The PVC status signaling works the same way as in the like-to-like case. The PE router reports the PVC status to the CE router based on the availability of the other CE router.
- Only FR DLCI mode is supported. FR port mode is not supported.
- If the Ethernet frame includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame and it is forwarded to the FR CE (Figure 18-9).

Figure 18-9 Protocol Stack for FR to Ethernet Local Switching Bridged Interworking—With VLAN Header



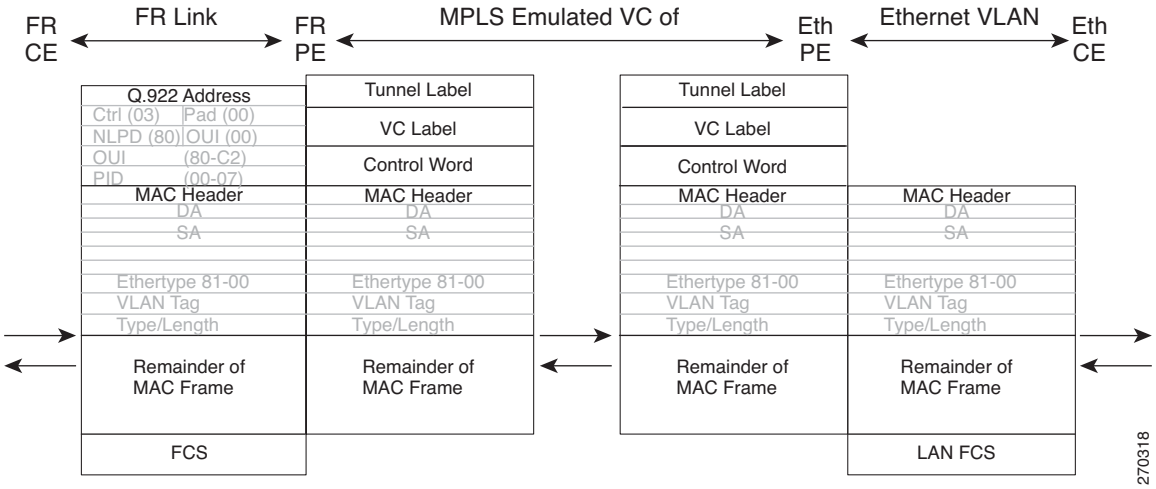
In Cisco IOS Release 12.2(33)SB, the Ethernet/VLAN to Frame Relay AToM has the following restrictions:

- The following translations are only supported and other translations are dropped:
  - Ethernet without LAN FCS (0300800080C20007)

  - Spanning tree (0300800080C2000E)

- The PE router automatically supports translation of both Cisco and IETF FR encapsulation types coming from the CE, but translates only to IETF when sending to the CE router. This is not a problem for the Cisco CE router, because it can manage IETF encapsulation upon receipt even if it is configured to send a Cisco encapsulation.
- The PVC status signaling works the same way as in the like-to-like case. The PE router reports the PVC status to the CE router based upon the availability of the pseudowire.
- The attachment circuit maximum transmission unit (MTU) must match when connected over MPLS.
- Only FR DLCI mode is supported. FR port mode is not supported.
- If the Ethernet frame includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame across the pseudowire (Figure 18-10).
- FR encapsulation types supported for routed interworking are Cisco and IETF for incoming traffic. However, IETF is also supported for outgoing traffic traveling to the CE only.

Figure 18-10 Protocol Stack for FR to Ethernet AToM Bridged Interworking—With VLAN Header



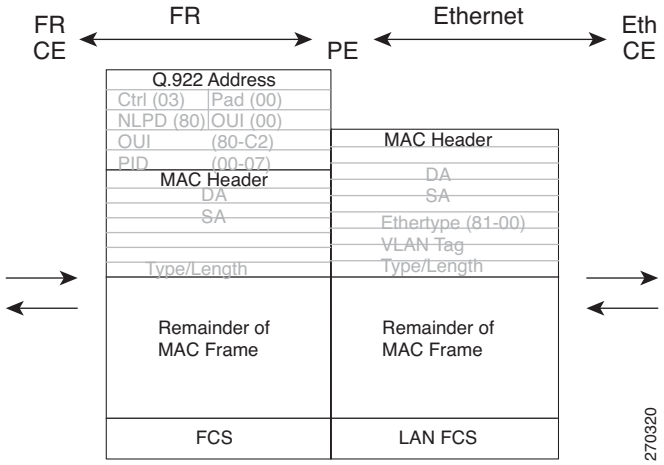
FR DLCI to Ethernet Local Switching—Bridged Interworking

This interworking type provides interoperability between Frame Relay attachment VC and Ethernet attachment VC connected to the same PE router. For this interworking type, Bridged encapsulation is used, corresponding to Bridged (Ethernet) Interworking mechanism.

- In the Ethernet to FR direction, the PE router forwards the Layer 2 packet without any change to the egress interface, encapsulating the L2 packet over FR using Bridged encapsulation.
- In the FR to Ethernet direction, the FR header and bridged encapsulation are discarded and the L2 packet is sent out with Ethernet encapsulation.

Figure 18-11 shows the protocol stack for FR to Ethernet local switching (bridged interworking).

Figure 18-11 Protocol Stack for FR to Ethernet Local Switching Bridged Interworking



The PE router automatically supports translation of both Cisco and IETF FR encapsulation types traveling from the CE, but translates only to IETF when sending to the CE router. This is not a problem for the Cisco CE router, because it can manage IETF encapsulation on receipt even if it is configured to send a Cisco encapsulation.

FR DLCI to VLAN 802.1Q Local Switching—Bridged Interworking

This interworking type provides interoperability between Frame Relay Attachment VC and Ethernet VLAN Attachment VC connected to the same PE router. For this interworking type the Bridged Encapsulation is used, corresponding to Bridged (Ethernet) Interworking mechanism.

In the case of an Ethernet VLAN attachment, the VLAN ID is a service delimiter, so the VLAN header is not included in the frame to or from the FR CE.

- In the VLAN to FR direction, the PE router discards the VLAN header from the Layer 2 packet. The PE router sends the frame to the FR egress interface after encapsulating the L2 packet over FR using Bridged encapsulation.
- In the FR to VLAN direction, the FR header and bridged encapsulation are discarded and the L2 packet is sent out with a VLAN header inserted, followed by the destination/source MAC addresses.

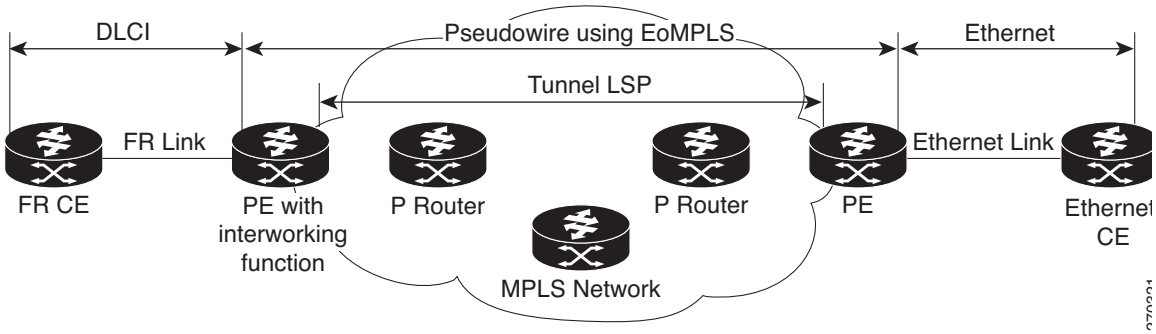
The protocol stack for FR to Ethernet local switching (bridged interworking) is shown in Figure 18-11.

FR DLCI to Ethernet Port AToM—Bridged Interworking

This interworking type provides interoperability between FR attachment VC and Ethernet attachment VC connected to different PE routers. Bridged encapsulation is used, corresponding to the Bridged (Ethernet) Interworking mechanism.

For an FR to Ethernet Port case, the interworking function is performed at the PE connected to the FR attachment VC based on multiprotocol interconnect over Frame Relay (Figure 18-12). The Interworking is implemented similar to an ATM-to-Ethernet case.

Figure 18-12 Network Topology for FR to Ethernet AToM Bridged Interworking



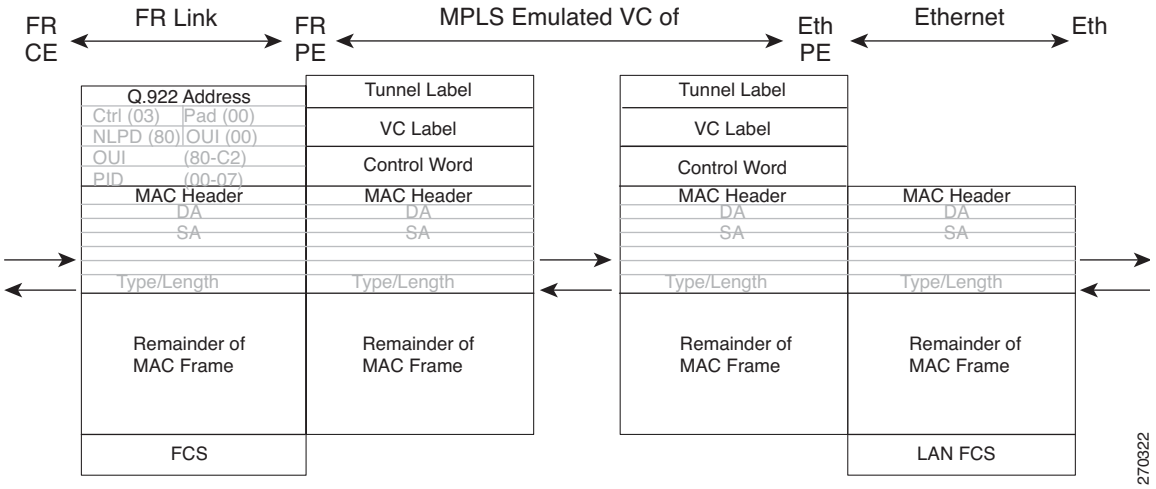
The advantage of this architecture is that the Ethernet PE (connected to the Ethernet segment) operates similarly to Ethernet like-to-like services: a pseudowire label is assigned to the Ethernet port and then the remote Label Distribution Protocol (LDP) session distributes the labels to its peer PE. Ethernet frames are carried through the MPLS network using Ethernet over MPLS (EoMPLS).

On the PE with Interworking function, in the direction from the FR segment to MPLS cloud, the bridged encapsulation (FR/SNAP header) is discarded and the Ethernet frame is encapsulated with the labels required to go through the pseudowire using the VC type 5 (Ethernet) (Figure 18-13).

In the opposite direction, after the label disposition from the MPLS cloud, Ethernet frames are encapsulated over FR using bridged encapsulation.

The Figure 18-13 shows the protocol stack for FR to Ethernet Bridged Interworking.

Figure 18-13 Protocol Stack for FR to Ethernet AToM Bridged Interworking—Without VLAN Header



FR DLCI to Ethernet VLAN 802.1Q AToM—Bridged Interworking

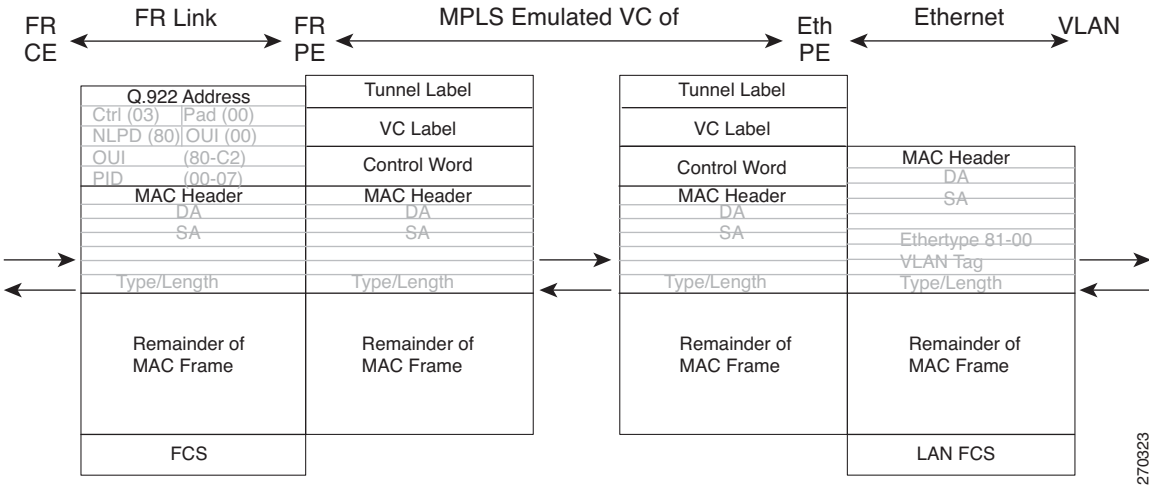
This interworking type provides interoperability between FR attachment VC and Ethernet VLAN Attachment VC connected to different PE routers. The bridged encapsulation is used, corresponding to the Bridged (Ethernet) Interworking mechanism.

The interworking function is performed in the same way as for FR to Ethernet port case, implemented on the PE connected to the FR attachment VC, based upon a multiprotocol interconnect over Frame Relay (see Figure 18-13).

As in the ATM to VLAN case, one difference exists on Ethernet side due the existence of the VLAN header in the incoming packet. The PE on the VLAN side discards the VLAN header of the incoming frames from the VLAN CE, and the PE inserts a VLAN header into the Ethernet frames traveling from the MPLS cloud. The frames sent on the pseudowire (with VC type 5) are Ethernet frames without the VLAN header.

The Figure 18-14 shows the protocol stack for FR to VLAN AToM Bridged Interworking.

Figure 18-14 Protocol Stack for FR to VLAN AToM Bridged Interworking



Configuration Tasks and Examples

This section describes configuration tasks for and examples of two L2VPN technology solutions

- Local Switching
- AToM

Local Switching

Figure 18-7 on page 18-9 shows LS configurations. Note that LS interworking in the Cisco 10000 router only supports the bridged interworking function, also known as Ethernet interworking function.

This section explains the following LS configurations and provides examples:

- FR DLCI to Ethernet Port, page 18-18
- FR DLCI to Ethernet VLAN 802.1Q, page 18-19

FR DLCI to Ethernet Port

You can configure the FR DLCI to Ethernet port feature on a router using the following steps:

1. **config t**
2. **frame-relay switching**
3. **interface serial** slot/subslot/port[:channel | .channel]
4. **encapsulation frame-relay**
5. **frame-relay intf-type dce**

6. **frame-relay interface-dlci** DLCI **switched**
7. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port
8. **no ip address**
9. **connect** connection-name [**fastethernet** | **gigabitethernet**] slot/subslot/port **serial** slot/subslot/port[:channel | .channel] **interworking ethernet**

Note The order of the interfaces in the **connect** command is not important.

The following example shows how you can configure the FR DLCI to Ethernet Port feature on a router:

```
config t
frame-relay switching
interface serial 2/0/0:1
    encapsulation frame-relay
    frame-relay intf-type dce
    frame-relay interface-dlci 100 switched
interface gigabitethernet 5/1/0
    no ip address
connect atm-enet gigabitethernet 5/1/0 serial 2/0/0:1 100 interworking ethernet
```

FR DLCI to Ethernet VLAN 802.1Q

You can configure the FR DLCI to Ethernet VLAN 802.1Q feature on a router using the following steps:

1. **config t**
2. **frame-relay switching**
3. **interface serial** slot/subslot/port[:channel | .channel]
4. **encapsulation frame-relay**
5. **frame-relay intf-type dce**
6. f**rame-relay interface-dlci** DLCI **switched**
7. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface
8. **encapsulation dot1q** VLAN-ID
9. **connect** connection-name [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface **serial** slot/subslot/port[:channel | .channel] **interworking ethernet**

Note The order of the interfaces in the **connect** command is not important.

The following example shows how you can configure the FR DLCI to Ethernet VLAN 802.1Q feature on a router:

```
config t
frame-relay switching
interface serial 2/0/0:1
    encapsulation frame-relay
    frame-relay intf-type dce
    frame-relay interface-dlci 100 switched
interface gigabitethernet 5/1/0.3
    encapsulation dot1q 2
connect fr-vlan gigabitethernet 5/1/0.3 serial 2/0/0:1 100 interworking ethernet
```

AToM

Figure 18-8 on page 18-11 illustrates different AToM configurations. Note that AToM interworking in the Cisco 10000 router only supports the bridged interworking function, also known as Ethernet interworking function.

This section explains the following AToM configurations and provides examples:

- FR DLCI to Ethernet Port, page 18-20
- FR DLCI to Ethernet VLAN 802.1Q, page 18-21

FR DLCI to Ethernet Port

You can configure the FR DLCI to Ethernet port feature on a PE1 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class name**
6. **encapsulation mpls**
7. **interworking ethernet**
8. **frame-relay switching**
9. **interface serial** slot/subslot/port[:channel | .channel]
10. **encapsulation frame-relay**
11. **frame-relay interface-dlci** DLCI **switched**
12. **connect mpls serial** slot/subslot/port[:channel | .channel] DLCI **l2transport**
13. **xconnect** remote-ip-address vc-id **pw-class** name

You can configure the FR DLCI to Ethernet port feature on a PE2 router using the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class name**
6. **encapsulation mpls**
7. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port
8. **xconnect** remote-ip-address vc-id **pw-class** name

Note The PE2 configuration does not include the **interworking ethernet** command because it is treated as like-to-like, as the attachment circuit is already an Ethernet port.

The following example shows how to configure the FR DLCI to Ethernet port feature on a PE1 router:

```
config t
mpls label protocol ldp
interface Loopback100
```

```
 ip address 10.0.0.100 255.255.255.255
pseudowire-class fr-eth
 encapsulation mpls
 interworking ethernet
frame-relay switching
interface serial 2/0/0:1
 encapsulation frame-relay
 frame-relay intf-type dce
connect mpls serial 2/0/0:1 567 l2transport
 xconnect 10.0.0.200 150 pw-class fr-eth
```

The following example shows how to configure the FR DLCI to an Ethernet port feature on a PE2 router:

```
config t
mpls label protocol ldp
interface Loopback200
 ip address 10.0.0.200 255.255.255.255
pseudowire-class fr-eth
 encapsulation mpls
interface gigabitethernet 5/1/0
 xconnect 10.0.0.100 150 pw-class fr-eth
```

FR DLCI to Ethernet VLAN 802.1Q

To configure the FR DLCI to Ethernet VLAN 802.1Q feature on a PE1 router, use the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class name**
6. **encapsulation mpls**
7. **interworking ethernet**
8. **frame-relay switching**
9. **interface serial** slot/subslot/port[:channel | .channel]
10. **encapsulation frame-relay**
11. **frame-relay intf-type dce**
12. **frame-relay interface-dlci** DLCI **switched**
13. **connect mpls serial** slot/subslot/port[:channel | .channel] DLCI **l2transport**
14. **xconnect** remote-ip-address vc-id **pw-class** name

To configure the FR DLCI to Ethernet VLAN 802.1Q feature on a PE2 router, use the following steps:

1. **config t**
2. **mpls label protocol ldp**
3. **interface** Loopback<name>
4. **ip address** local-ip-address local-mask
5. **pseudowire-class** name
6. **encapsulation mpls**
7. **interworking ethernet**

8. **interface** [**fastethernet** | **gigabitethernet**] slot/subslot/port.subinterface
9. **encapsulation dot1q** VLAN-ID
10. **xconnect** remote-ip-address vc-id **pw-class** name

Note In the case of an FR DLCI to VLAN, the PE2 configuration includes the **interworking ethernet** command.

The following example shows how to configure the FR DLCI to Ethernet VLAN 802.1Q feature on a PE1 router:

```
config t
mpls label protocol ldp
interface Loopback100
 ip address 10.0.0.100 255.255.255.255
pseudowire-class fr-vlan
 encapsulation mpls
 interworking ethernet
frame-relay switching
interface serial 2/0/0:1
 encapsulation frame-relay
 frame-relay intf-type dce
connect mpls serial 2/0/0:1 567 l2transport
 xconnect 10.0.0.200 150 pw-class fr-vlan
```

The following example shows how to configure the FR DLCI to Ethernet VLAN 802.1Q feature on a PE2 router:

```
config t
mpls label protocol ldp
interface Loopback200
 ip address 10.0.0.200 255.255.255.255
pseudowire-class fr-vlan
 encapsulation mpls
 interworking ethernet
interface gigabitethernet 5/1/0.3
 encapsulation dot1q 1525
 xconnect 10.0.0.100 150 pw-class fr-vlan
```

Note To verify the L2VPN interworking status and check the statistics, refer to the "Verifying L2VPN Interworking" section on page 18-22.

Verifying L2VPN Interworking

To verify the L2VPN status - local switching, use the following commands:

- **show connection [all | name | id | elements | port]**
- **show pxf cpu atom [circuits | interface | vcci]**

To view the L2VPN statistics - local switching, use the following command:

- **show pxf cpu statistics atom**

To verify the L2VPN status - AToM, use the following commands:

- **show connection [all | name | id | elements | port]**
- **show xconnect [all | interface | peer]**

- **show mpls l2transport [binding | checkpoint | hw-capability | summary | vc]**
- **show mpls infrastructure lfd pseudowire vcid**
- **show pxf cpu atom [circuits | interface | vcci]**

To verify the L2VPN statistics - AToM, use the following commands:

- **show pxf cpu statistics atom**
- **show pxf cpu subblocks**

CHAPTER 19

Configuring Multilink Point-to-Point Protocol Connections

LAN-based applications and information transfer services, such as electronic mail, can transmit large amounts of traffic, placing increased demand on today's wide-area networks (WANs). The costs of starting and maintaining a WAN network are also on the rise. Therefore, a reliable and cost-effective solution is needed that makes efficient use of WAN links. Multilink Point-to-Point Protocol (MLP) is the solution implemented on the Cisco 10000 series router.

This chapter describes MLP and how to configure it on serial and ATM connections on the Cisco 10000 series router. It includes the following topics:

Multilink Point-to-Point Protocol

Multilink Point-to-Point Protocol (MLP) is used to combine multiple physical links into a single logical connection or MLP bundle (see Figure 19-1). Using MLP, you can increase bandwidth and more easily manage all of the circuits through a single interface. The MLP connection has a maximum bandwidth that is equal to the sum of the bandwidths of the component links. MLP also provides load balancing, multivendor interoperability, packet fragmentation and reassembly, and increased redundancy. The Cisco 10008 router implements the MLP specifications defined in RFC 1990.

MLP provides traffic load balancing over multiple wide-area network (WAN) links by sending packets and packet fragments over the links of bundle members. The multiple links come up in response to a defined load threshold. MLP mechanisms can calculate load on both inbound and outbound traffic, or on either direction as needed for the traffic between specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. Large nonreal-time packets are multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic. However, the smaller real-time packets are not multilink encapsulated. Instead, MLP interleaving provides a special transmit queue (priority queue) for these delay-sensitive packets to allow the packets to be sent earlier than other packet flows. Real-time packets remain intact and MLP interleaving mechanisms send the real-time packets between fragments of the larger nonreal-time packets. For more information about link fragmentation and interleaving, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

MLP can provide increased redundancy by allowing traffic to flow over the remaining member links when a port fails. You can configure the member links on separate physical ports on the same line card or on different line cards. If a port becomes unavailable, MLP directs traffic over the remaining member links with minimal disruption to the traffic flow.

MLP mechanisms preserve packet ordering over an entire bundle, guaranteeing that network packets are processed at the receiving system in the same order that they are logically transmitted.

Valid multilink interface values for MLP over serial or multi-VC MLP over ATM are from 1 to 9999 (Release 12.2(28)SB and later), or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). For example:

```
Router(config)# interface multilink 8
```

The Cisco 10008 router supports the following MLP features:

- MLP Over Serial Interfaces, page 19-13
- Single-VC MLP Over ATM Virtual Circuits, page 19-15
- Multi-VC MLP Over ATM Virtual Circuits, page 19-16
- MLP on LNS, page 19-18

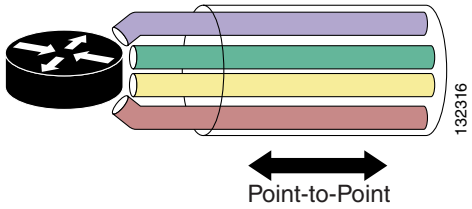
Feature History for Multilink PPP

Cisco IOS Release	Description	Required PRE
12.0(23)SX	The MLP over Serial feature was introduced on the Cisco 10000 series router.	PRE1
12.2(28)SB	The MLP over Serial, Single-VC MLP over ATM VCs, and Multi-VC MLP over ATM VCs features were introduced on the PRE2.	PRE2
12.2(31)SB2	Support was added for the PRE3 and the valid multilink interface ranges for MLP over serial or multi-VC MLP over ATM changed from 1 to 9999 (Release 12.2(28)SB and later) to from 1 to 9999 and 65,536 to 2,147,483,647.	PRE3
12.2(33)SB	The MLPPP on LNS feature has been introduced on the Cisco 10000 series router that is supported on PRE3 and PRE4. This feature is not supported on PRE2.	PRE3 and PRE4

MLP Bundles

MLP combines multiple physical links into a logical bundle called an MLP bundle (see Figure 19-1). an MLP bundle is a single, virtual interface that connects to the peer system. Having a single virtual interface enables fancy queuing and QoS to be applied to the traffic on the virtual interface (for example, policing and traffic shaping can be applied to the traffic flows). Each individual link to the peer system might be doing some form of fancy queuing, but none of the links knows about the traffic on the other parallel links. Fancy queuing and QoS cannot be applied uniformly to the entire aggregate traffic between the system and its peer system. A single virtual interface also simplifies the task of monitoring traffic to the peer system (for example, traffic statistics are all on one interface).

Figure 19-1 Multilink PPP Bundle



An endpoint discriminator is used to identify the member links of the MLP bundle.

Restrictions for MLP Bundles

The router supports links equal to T1/E1 or less for MLPPP bundling. You cannot bundle high speed links (for example, E3) because the router can store only 50 ms of data based on the E1 speed.

MLP Bundles and PPP Links

MLP works with fully functional Point-to-Point Protocol (PPP) interfaces. an MLP bundle can consist of a PPP over serial link and a PPP over ATM link. As long as each link behaves like a standard serial interface, the mixed links work properly in a bundle.

Adding the **ppp multilink group** command to a link's configuration does not make that link part of the specified bundle. This command only places a restriction on the link. If the link negotiates to use multilink, then it must provide the proper identification to join the bundle on the multilink interface or to activate a bundle on that interface. If the link provides identification that coincides with another active bundle in the system, or the link fails to match the identity of a bundle that is already active on the multilink group interface, the connection terminates.

A link joins an MLP bundle only if it negotiates to use multilink when the connection is established and the identification information exchanged matches that of an existing bundle. If a link supplies identification information that does not match any known bundle, MLP creates a new bundle for the user.

System Limits for MLP Bundles

Table 19-1 lists the system limits for MLP bundles.

Table 19-1 System Limits for MLP Bundles

Feature	Maximum No. of Members Per Bundle	Maximum No. of Bundles Per System	Maximum No. of Member Links Per System	Multilink Interface Range	LFI Supported
MLP over Serial	10	1250	2500	1 to 9999 (Release 12.2(28)SB and later) and from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later)	Yes Interleaving on all member links
Single-VC MLP over ATM	1	8192	8192	10,000 and higher	Yes Interleaving on 1 member link
Multi-VC MLP over ATM	10	1250	2500	1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later)	Yes Interleaving on 1 member link

Note The multilink interface ranges described in Table 19-1 require Cisco IOS Release 12.2(28)SB or later releases. For releases earlier than to Cisco IOS Release 12.2(28)SB, the valid multilink interface range is 1 to 2,147,483,647.

Types of MLP Bundle Interfaces

MLP bundle interfaces can be one of the following:

- Virtual access interfaces (VAIs)
- Multilink group interfaces

Both of these interfaces provide the same level of PPP and multilink functionality once a bundle is established, and all PPP and multilink-related features run identically on the bundle.

A VAI is the primary type of interface used for MLP bundles. It is created dynamically for a multilink connection and released as soon as the connection is torn down. A bundle interface of this type exists only as long as a user is connected. As soon as the user disconnects, the bundle interface no longer exists. VAIs are the default type of bundle interface. If you do not configure a multilink group interface, the bundle interface is automatically a VAI, which has the following advantages and disadvantages:

- The number of bundle interfaces depends only on the number of currently active multilink users and not on the size of your user database.
- Because a local configuration source does not exist for per-user information, this information is derived from another source, such as an AAA (authentication, authorization, and accounting) server.
- Because a dedicated interface does not exist for you to monitor, you must track a user's activity using another means, such as the accounting mechanism of an AAA server.

Note Cisco 10000 series routers does not support VAI in a back-to-back connection. VAI is supported only at LNS (MLPoLNS).

Multilink group interfaces are static interfaces that exist whether or not they are being used at a particular point in time. Multilink group interfaces are dedicated to specific remote users and are primarily used in leased-line environments in which you already know where all of your physical links are connected and the number of users is primarily defined by the number of physical connections your system has.

Multilink group interfaces allow you to track a specific user's activity. By examining a user's associated interface, you can easily see if a user is connected and how much traffic the user has sent or received. You can monitor the state of the multilink group interface for such things as network outages.

MLP Groups

When you configure the **ppp multilink group** command on a link, the command applies a restriction to the link that indicates the link is not allowed to join any bundle other than the indicated group interface, and that the connection is to be terminated if the peer system attempts to join a different bundle.

A link actually joins a bundle when the identification keys for that link match the identification keys for an existing bundle (see the "How MLP Determines the Link a Bundle Joins" section on page 19-6). Configuring the **ppp multilink group** command on a link does not allow the link to bypass this process, unless a bundle does not already exist for this particular user. When matching links to bundles, the identification keys are always the determining factors.

Because the **ppp multilink group** command merely places a restriction on the link, any MLP-enabled link that is not assigned to a particular multilink group can join the dedicated bundle interface if it provides the correct identification keys for that dedicated bundle. Removing the **ppp multilink group** command from an active link that currently is a member of a multilink group does not make that link leave the bundle because the link is still a valid member. It is just no longer restricted to this one bundle.

MLP Group Interfaces and Virtual Template Interfaces

You can configure MLP by assigning a multilink group to a virtual template interface configuration. Virtual templates allow a virtual access interface (VAI) to dynamically clone interface parameters from the specified virtual template. If you assign a multilink group to a virtual template and you assign the virtual template to a physical interface, all of the links that pass through the physical interface belong to the same multilink bundle.

A multilink group interface configuration overrides a global multilink virtual template configured using the **multilink virtual template** command.

On the Cisco 10008 router, you can use multilink group interfaces with ATM and serial interfaces. To configure MLP using a multilink group interface, do the following:

- Configure the multilink group under the ATM PVC or any other interface
- Assign the ppp multilink to a virtual template
- Configure the physical interface to use the virtual template.

For more information, see the "Changing the Default Endpoint Discriminator" section on page 19-34.

How MLP Determines the Link a Bundle Joins

A link joins a bundle when the identification keys for that link match the identification keys for an existing bundle.

Two keys define the identity of a remote system: the PPP username and the MLP endpoint discriminator. The PPP authentication mechanisms (for example, PAP or CHAP) learn the PPP username. The endpoint discriminator is an option negotiated by the Link Control Protocol (LCP). Therefore, a bundle consists of all of the links that have the same PPP username and endpoint discriminator.

A link that does not provide a PPP username or endpoint discriminator is an anonymous link. MLP collects all of the anonymous links into a single bundle referred to as the anonymous bundle or default bundle. Typically, there can be only one anonymous bundle. Any anonymous links that negotiate MLP join (or create) the anonymous bundle.

When using multilink group interfaces, more than one anonymous peer is allowed. When you pre-assign a link to an MLP bundle by using the **ppp multilink group** command, and the link is anonymous, the link joins the bundle interface it is assigned to if the interface is not already active and associated with a non-anonymous user.

MLP determines the bundle a link joins in the following way:

- When a link connects, MLP creates a bundle name identifier for the link.
- MLP then searches for a bundle with the same bundle name identifier.
  - If a bundle with the same identifier exists, the link joins that bundle.
  - If a bundle with the same identifier does not exist, MLP creates a new bundle with the same identifier as the link, and the link is the first link in the bundle.

Table 19-2 describes the commands and associated algorithm used to generate a bundle name. In the table, username typically means the authenticated username; however, an alternate name can be used instead. The alternate name is usually an expanded version of the username (for example, VPDN tunnels might include the network access server name) or a name derived from other sources.

Table 19-2 Bundle Name Generation

Command	Bundle Name Generation Algorithm
multilink bundle-name authenticated	The bundle name is the peer's username, if available. If the peer does not provide a username, the algorithm uses the peer's endpoint discriminator. **Note** The **authenticated** keyword specifies that the bundle name is based on whatever notion of a username the system can derive. The endpoint discriminator is ignored entirely, unless it is the only name that can be found. The **multilink bundle-name authenticated** command is the default naming policy.
multilink bundle-name endpoint	The bundle name is the peer's endpoint discriminator. If there is no endpoint discriminator, the algorithm uses the peer's username.
multilink bundle-name both	The name of the bundle is a concatenation of the username and the endpoint discriminator.

IP Addresses on MLP-Enabled Links

Configuring an IP address on a link used for MLP does not always work as expected. For example, consider the following configuration:

```
interface Serial 1/0/0
    ip address 10.2.3.4 255.255.255.0
    encapsulation ppp
    ppp multilink
```

You might expect the following behavior as a result of the above configuration:

- If the interface does not negotiate to use MLP and the interface comes up as a regular PPP link, then the interface negotiates the Internet Protocol Control Protocol (IPCP) and its local address is 10.2.3.4.
- If the interface did negotiate to use MLP, then the configured IP address is meaningless because the link is not visible to IP while it is part of a bundle. The bundle is a network level interface and can have its own IP address, depending on the configuration used for the bundle.

However, configuring an IP address on an MLP-enabled link does not work as you might expect. Instead, if a link with an IP address configured comes up and joins a bundle, IP installs a route directly to that link interface and it might try to route packets directly to that link, bypassing the MLP bundle. This occurs because IP considers an interface to be up for IP traffic whenever IP is configured on the interface and the interface is up. MLP intercepts and discards these misdirected frames. This condition occurs frequently if you use a virtual template interface to configure both the PPPoX member links and the bundle interface.

Using unnumbered IP interfaces enables you to work around IP problems and configure an IP address on an MLP-enabled link. For example, the following is a sample configuration for Multi-VC MLP over ATM using an unnumbered IP interface:

```
!
interface Multilink1
    ip unnumbered Loopback0
    peer default ip address pool mlpoa_pool
    ppp chap hostname m1
    ppp multilink
    ppp multilink group 1
!
interface atm 2/0/0
    no ip address
!
interface atm 2/0/0.1 point-to-point
    pvc 0/32
        ppp multilink group 1
        vbr-nrt 128 64 20
        encapsulation aal5mux ppp Virtual-Template1
!
!
interface atm 2/0/0.2 point-to-point
    pvc 0/33
        ppp multilink group 1
        vbr-nrt 128 64 20
        encapsulation aal5mux ppp Virtual-Template1
!
interface Virtual-Template1
    no ip address
    keepalive 30
    ppp max-configure 110
    ppp max-failure 100
    ppp multilink
    ppp timeout retry 5
!
ip local pool mlpoa_pool 100.1.1.1 100.1.7.254
!
```

Valid Ranges for MLP Interfaces

Table 19-3 lists the valid ranges you can specify when creating MLP interfaces using the **interface multilink** command.

Table 19-3 MLP Interface Ranges

Cisco IOS Release	PRE2 MLP Interface Ranges	PRE3 MLP Interface Ranges
Release 12.2(28)SB and later	1 to 9999	Not Applicable
Release 12.2(31)SB2 and later	1 to 9999 65,536 to 2,147,483,647	1 to 9999 65,536 to 2,147,483,647

MLP Overhead

MLP encapsulation adds six extra bytes (4 header, 2 checksum) to each outbound packet. These overhead bytes reduce the effective bandwidth on the connection; therefore, the throughput for an MLP bundle is slightly less than an equivalent bandwidth connection that is not using MLP. If the average packet size is large, the extra MLP overhead is not readily apparent; however, if the average packet size is small, the extra overhead becomes more noticeable.

Using MLP fragmentation adds additional overhead to a packet. Each fragment contains six bytes of MLP header plus a link encapsulation header (for example, a High Level Data Link Control (HDLC) header).

Configuration Commands for MLP

This section describes the following commands used to configure MLP and MLP-based link fragmentation and interleaving:

- interface multilink Command, page 19-9
- ppp multilink Command, page 19-10
- ppp multilink fragment-delay Command, page 19-10
- ppp multilink interleave Command, page 19-11
- ppp multilink fragment disable Command, page 19-12
- ppp multilink fragment disable Command, page 19-12
- ppp multilink group Command, page 19-12

For more information about MLP-based link fragmentation and interleaving, see the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

interface multilink Command

To create and configure a multilink bundle, use the **interface multilink** command in global configuration mode. To remove a multilink bundle, use the **no** form of the command.

interface multilink *multilink-bundle-number*

no interface multilink *multilink-bundle-number*

Syntax Description

multilink-bundle-number	A nonzero number that identifies the multilink bundle.

Command History

Cisco IOS Release	Description
12.0	The **interface multilink** command was introduced on the Cisco 10000 series router.
12.2(16)BX	This command was introduced on the PRE2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(31)SB2	This command was introduced on the PRE3. Valid multilink interface values changed. See the Usage Guides or Table 19-3 on page 19-8.

Defaults No multilink interfaces are configured.

Usage Guidelines For Cisco IOS Release 12.2(28)SB and later releases, the range of valid values for multilink interfaces are the following:

- MLP over Serial—1 to 9999 (Release 12.2(28)SB and later), and 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later)
- Single-VC MLP over ATM—10,000 and higher
- Multi-VC MLP over ATM—1 to 9999 (Release 12.2(28)SB and later), and 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later)

For releases earlier than to Cisco IOS Release 12.2(28)SB, the valid multilink interface range is 1 to 2,147,483,647.

ppp multilink Command

To enable Multilink PPP (MLP) on an interface, use the **ppp multilink** command in interface configuration mode. To disable MLP, use the **no** form of the command.

ppp multilink

no ppp multilink

Command History

Cisco IOS Release	Description
12.0(23)SX	The **ppp multilink** command was introduced on the Cisco 10000 series router.
12.2(16)BX	This command was introduced on the PRE2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Defaults The command is disabled.

Usage Guidelines The **ppp multilink** command applies only to interfaces that use Point-to-Point Protocol (PPP) encapsulation.

When you use the **ppp multilink** command, the first channel negotiates the appropriate Network Control Protocol (NCP) layers (such as the IP Control Protocol and IPX Control Protocol), but subsequent links negotiate only the Link Control Protocol (LCP) and MLP.

ppp multilink fragment-delay Command

To specify a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle, use the **ppp multilink fragment-delay** command in interface configuration mode. To reset the maximum delay to the default value, use the **no** form of the command.

ppp multilink fragment-delay ***delay-max***

no ppp multilink fragment-delay ***delay-max***

Syntax Description

delay-max	Specifies the maximum amount of time, in milliseconds, that is required to transmit a fragment. Valid values are from 1 to 1000 milliseconds.

Command History

Cisco IOS Release	Description
12.0(23)SX	The **ppp multilink fragment-delay** command was introduced on the Cisco 10000 series router.
12.2(16)BX	This command was introduced on the PRE2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Defaults

If fragmentation is enabled, the fragment delay is 30 milliseconds.

Usage Guidelines

The **ppp multilink fragment-delay** command is useful when packets are interleaved and traffic characteristics such as delay, jitter, and load balancing must be tightly controlled.

MLP chooses a fragment size on the basis of the maximum delay allowed. If real-time traffic requires a certain maximum boundary on delay, using the **ppp multilink fragment-delay** command to set that maximum time can ensure that a real-time packet gets interleaved within the fragments of a large packet.

By default, MLP has no fragment size constraint, but the maximum number of fragments is constrained by the number of links. If interleaving is enabled, or if a fragment delay is explicitly configured with the **ppp multilink fragment-delay** command, then MLP uses a different fragmentation algorithm. In this mode, the number of fragments is unconstrained, but the size of each fragment is limited to the fragment-delay value, or 30 milliseconds if the fragment delay has not been configured.

The **ppp multilink fragment-delay** command is configured under the multilink interface. The value assigned to the *delay-max* argument is scaled by the speed at which a link can convert the time value into a byte value.

ppp multilink interleave Command

To enable interleaving of real-time packets among the fragments of larger nonreal-time packets on a Multilink PPP (MLP) bundle, use the **ppp multilink interleave** command in interface configuration mode. To disable interleaving, use the **no** form of the command.

ppp multilink interleave

no ppp multilink interleave

Command History

Cisco IOS Release	Description
12.0(23)SX	The **ppp multilink interleave** command was introduced on the Cisco 10000 series router.
12.2(16)BX	This command was introduced on the PRE2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Defaults

Interleaving is disabled.

Usage Guidelines The **ppp multilink interleave** command applies to multilink interfaces, which are used to configure a bundle.

Interleaving works only when the queuing mode on the bundle is set to fair queuing.

If interleaving is enabled when fragment delay is not configured, the default delay is 30 milliseconds. The fragment size is derived from that delay, depending on the bandwidths of the links.

ppp multilink fragment disable Command

To disable packet fragmentation, use the **ppp multilink fragment disable** command in interface configuration mode. To enable fragmentation, use the **no** form of this command.

ppp multilink fragment disable

no ppp multilink fragment disable

Command History

Cisco IOS Release	Description
11.3	This command was introduced as **ppp multilink fragmentation**.
12.2	The **no ppp multilink fragmentation** command was changed to **ppp multilink fragment disable**. The **no ppp multilink fragmentation** command was recognized and accepted through Cisco IOS Release 12.2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines The **ppp multilink fragment delay** and **ppp multilink interleave** commands have precedence over the **ppp multilink fragment disable** command. Therefore, the **ppp multilink fragment disable** command has no effect if these commands are configured for a multilink interface and the following message displays:

```
Warning: 'ppp multilink fragment disable' or 'ppp multilink fragment maximum' will be
ignored, since multilink interleaving or fragment delay has been configured and have
higher precedence.
```

To completely disable fragmentation, you must do the following:

```
Router(config-if)# no ppp multilink fragment delay
Router(config-if)# no ppp multilink interleave
Router(config-if)# ppp multilink fragment disable
```

ppp multilink group Command

To restrict a physical link to joining only a designated multilink group interface, use the **ppp multilink group** command in interface configuration mode. To remove the restriction, use the **no** form of the command.

ppp multilink group ***group-number***

no ppp multilink group ***group-number***

Syntax Description

group-number	Identifies the multilink group. This number must be identical to the *multilink-bundle-number* you assigned to the multilink interface. Valid values are: • MLP over Serial—1 to 9999 • Single-VC MLP over ATM—10,000 and higher • Multi-VC MLP over ATM—1 to 9999

Command History

Cisco IOS Release	Description
12.0	The **multilink-group** command was introduced on the Cisco 10000 series router.
12.2	This command was changed to **ppp multilink group**. The **multilink-group** command is accepted by the command line interpreter through Cisco IOS Release 12.2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Defaults

Disabled

Usage Guidelines

By default the **ppp multilink group** command is disabled, which means the link can negotiate to join any bundle in the system.

When the **ppp multilink group** command is configured, the physical link is restricted from joining any but the designated multilink group interface. If a peer at the other end of the link tries to join a different bundle, the connection is severed. This restriction applies when Multilink PPP (MLP) is negotiated between the local end and the peer system. The link can still come up as a regular PPP interface.

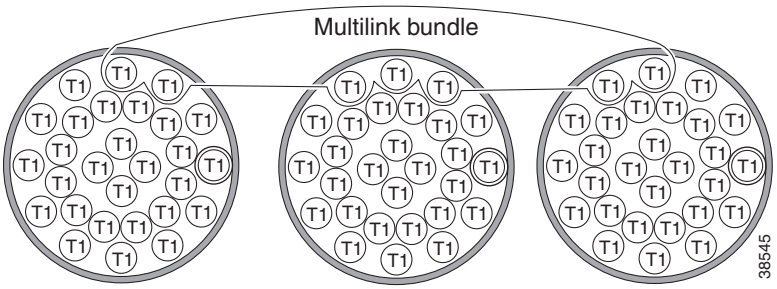
MLP Over Serial Interfaces

The MLP over Serial interfaces feature enables you to bundle together T1 interfaces into a single logical connection called an MLP bundle (see the "MLP Bundles" section on page 19-3). MLP over Serial also provides the following:

- Load balancing—MLP provides bandwidth on demand and uses load balancing across all member links (up to 10) to transmit packets and packet fragments. MLP mechanisms calculate the load on either the inbound or outbound traffic between specific sites. Because MLP splits packets and fragments across all member links during transmission, MLP reduces transmission latency across WAN links.
- Increased redundancy—MLP allows traffic to flow over the remaining member links when a port fails. By configuring an MLP bundle that consists of T1 lines from more than one line card, if one line card stops operating, the part of the bundle on the other line cards continues to operate.
- Link fragmentation and interleaving—The MLP fragmenting mechanism fragments large nonreal-time packets and sends the fragments at the same time over multiple point-to-point links to the same remote address. Smaller real-time packets remain intact. The MLP interleaving mechanism sends the real-time packets between the fragments of the nonreal-time packets, thus reducing real-time packet delay. For more information about link fragmentation and interleaving, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Figure 19-2 shows an MLP bundle that consists of T1 interfaces from three T3 interfaces.

Figure 19-2 MLP Bundle for Multilink PPP Over Serial Connections



Performance and Scalability for MLP Over Serial Interfaces

Configure the **hold-queue** command in interface configuration mode for all physical interfaces. For example:

```
Router(config-if)# hold-queue 4096 in
```

For more information, see the Scalability and Performance chapter in this guide.

Restrictions and Limitations for MLP Over Serial Interfaces

- A multilink bundle can have up to 10 member links. The router supports both full T1 interfaces and fractional T1 interfaces as member links, but fractional T1 interfaces are supported only when LFI is enabled.

Note You can terminate the serial links on multiple line cards in the router chassis if all of the links are the same type, such as T1 or E1.

- The router supports a maximum of 1250 bundles per system and a maximum of 2500 member links per system.
- The valid multilink interface ranges are from 1 to 9999 (Release 12.2(28)SB and later) and from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). For example:

```
Router(config)# interface multilink 8
```

- Interleaving is supported on all member links. MLP over Serial-based LFI must be enabled on an interface that has interleaving turned on.
- All member links in an MLP bundle must have the same encapsulation type and bandwidth.
- If a virtual template attached to a member link specifies a bandwidth, the router does not clone the specified bandwidth to the MLP bundle and the member links.
- You cannot manually configure the bandwidth on a bundle interface by using the **bandwidth** command.
- You cannot apply a virtual template with MLP configured to an MLP bundle.

- We strongly recommend that you use only strict priority queues when configuring MLP over Serial-based LFI. For more information, see the "Prioritizing Services" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Single-VC MLP Over ATM Virtual Circuits

The Single-VC MLP over ATM virtual circuits (VCs) feature enhances the MLP over Serial interfaces feature by enabling you to configure Multilink Point-to-Point Protocol (MLP) on an ATM VC. By doing so, you can aggregate multiple data paths (for example, PPP over ATM encapsulated ATM VCs) into a single logical connection called an MLP bundle (see the "MLP Bundles" section on page 19-3). The MLP bundle can have only one member link.

MLP supports link fragmentation and interleaving (LFI). When enabled, the MLP fragmentation mechanism multilink encapsulates large nonreal-time packets and fragments them into a small enough size to satisfy the delay requirements of real-time traffic. The smaller real-time packets remain intact and MLP sends the packets to a special transmit queue, allowing the packets to be sent earlier than other packet flows. The MLP interleaving mechanism sends the real-time packets between the fragments of the nonreal-time packets. For more information about link fragmentation and interleaving, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Performance and Scalability for Single-VC MLP Over ATM

- Configure the **hold-queue** command in interface configuration mode for all physical interfaces, except when configuring the OC-12 ATM line card. The 1-Port OC-12 ATM line card does not require the **hold-queue** command, for example:

```
Router(config-if)# hold-queue 4096 in
```

- Configure the following commands and recommended values on the virtual template interface:
  - ppp max-configure 110
  - ppp max-failure 100
  - ppp timeout retry 5
  - keepalive 30

  For example:

```
Router(config-if)# ppp max-configure 110
Router(config-if)# ppp max-failure 100
Router(config-if)# ppp timeout retry 5
Router(config-if)# keepalive 30
```

For more information, see the "Scalability and Performance" chapter in this guide.

Restrictions and Limitations for Single-VC MLP Over ATM

- Only one member link is supported per bundle.
- Single-VC MLP over ATM member links are restricted to non-aggregated PVCs (for example, variable bit rate-nonreal-time (VBR-nrt) and constant bit rate (CBR) ATM traffic classes only).
- The router supports a maximum of 8192 bundles per system and 8192 member links per system.

- Each member link can have a bandwidth rate up to 2048 kbps.
- The router only supports member links with the same encapsulation type.
- MLP PVCs cannot be on-demand VCs that are automatically provisioned.
- Associating MLP over ATM PVCs with ATM virtual paths (VPs) is discouraged, though not prevented.
- The valid multilink interface values are 10,000 to 65534, for example:

  ```
  Router(config)# interface multilink 10004
  ```

  The values higher than 65534 are used for multi-member bundle
- Cisco IOS software supports a maximum of 4096 total virtual template interfaces.
- You cannot manually configure the bandwidth on a bundle interface using the **bandwidth** command.
- If a virtual template attached to a member link specifies a bandwidth, the router does not clone the specified bandwidth to the MLP bundle and the member links.
- You cannot apply a virtual template with MLP configured to an MLP bundle.
- If link fragmentation and interleaving (LFI) is enabled, only one link is used for interleaving. For more information, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.
- We strongly recommend that you use only strict priority queues when configuring MLP over ATM-based LFI. For more information, see the "Prioritizing Services" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Multi-VC MLP Over ATM Virtual Circuits

The Multi-VC MLP over ATM virtual circuits (VCs) feature enhances the MLP over Serial interfaces feature by enabling you to configure Multilink Point-to-Point Protocol (MLP) on multiple ATM VCs. By doing so, you can aggregate multiple data paths (for example, PPP over ATM encapsulated ATM VCs) into a single logical connection called an MLP bundle (see the "MLP Bundles" section on page 19-3). an MLP bundle can have up to 10 member links.

Multi-VC MLP over ATM provides the following:

- Load balancing—MLP provides bandwidth on demand and uses load balancing across all member links (up to 10) to transmit packets and packet fragments. The multiple links come up in response to a defined load threshold. MLP mechanisms calculate load on both inbound and outbound traffic, or on either direction as needed for traffic between specific sites. Because MLP uses all member links to transmit packets and fragments, MLP reduces transmission latency across WAN links.
- Increased redundancy—MLP allows traffic to flow over the remaining member links when a port fails. You can configure the member links on separate physical ports on the same line card or on different line cards. If a port becomes unavailable, MLP directs traffic over the remaining member links with minimal disruption to the traffic flow. MLP mechanisms preserve packet ordering over an entire bundle.
- Link fragmentation and interleaving—The MLP fragmentation mechanism fragments packets and sends the fragments at the same time over multiple point-to-point links to the same remote address. MLP multilink encapsulates large nonreal-time packets and fragments them into a small enough size to satisfy the delay requirements of real-time traffic. The smaller real-time packets remain intact and

MLP sends the packets to a special transmit queue, allowing the packets to be sent earlier than other packet flows. The MLP interleaving mechanism sends the real-time packets between the fragments of the nonreal-time packets.

For more information about link fragmentation and interleaving, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Performance and Scalability for Multi-VC MLP Over ATM VCs

- Configure the **hold-queue** command in interface configuration mode for all physical interfaces, except when configuring the OC-12 ATM line card. The 1-Port OC-12 ATM line card does not require the **hold-queue** command, for example:

  ```
  Router(config-if)# hold-queue 4096 in
  ```

- Configure the following commands and recommended values on the virtual template interface:
  - ppp max-configure 110
  - ppp max-failure 100
  - ppp timeout retry 5
  - keepalive 30

  For example:

  ```
  Router(config-if)# ppp max-configure 110
  Router(config-if)# ppp max-failure 100
  Router(config-if)# ppp timeout retry 5
  Router(config-if)# keepalive 30
  ```

For more information, see the Scalability and Performance chapter in this guide.

Restrictions and Limitations for Multi-VC MLP Over ATM VCs

- A maximum of 10 member links is supported per bundle.
- MLP over ATM member links are restricted to non-aggregated PVCs (for example, variable bit rate-nonreal-time (VBR-nrt) and constant bit rate (CBR) ATM traffic classes only).
- The router supports a maximum of 1250 bundles per system and 2500 member links per system.
- Each member link can have a bandwidth rate up to 2048 kbps.
- The router only supports member links with the same encapsulation type.
- The valid multilink interface ranges are from 1 to 9999 (Release 12.2(28)SB and later) and from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). For example:

  ```
  Router(config)# interface multilink 8
  ```

- MLP PVCs cannot be on-demand VCs that are automatically provisioned.
- Associating MLP over ATM PVCs with ATM virtual paths (VPs) is discouraged, though not prevented.
- Cisco IOS software supports a maximum of 4096 total virtual template interfaces.
- You cannot manually configure the bandwidth on a bundle interface using the **bandwidth** command.

- You cannot apply a virtual template with MLP configured to an MLP bundle.
- If a virtual template attached to a member link specifies a bandwidth, the router does not clone the specified bandwidth to the MLP bundle and the member links.
- If link fragmentation and interleaving (LFI) is enabled, only one link is used for interleaving. For more information, see the "Fragmenting and Interleaving Real-Time and Nonreal-Time Packets" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.
- We strongly recommend that you use only strict priority queues when configuring Multi-VC MLP over ATM-based LFI. For more information, see the "Prioritizing Services" chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

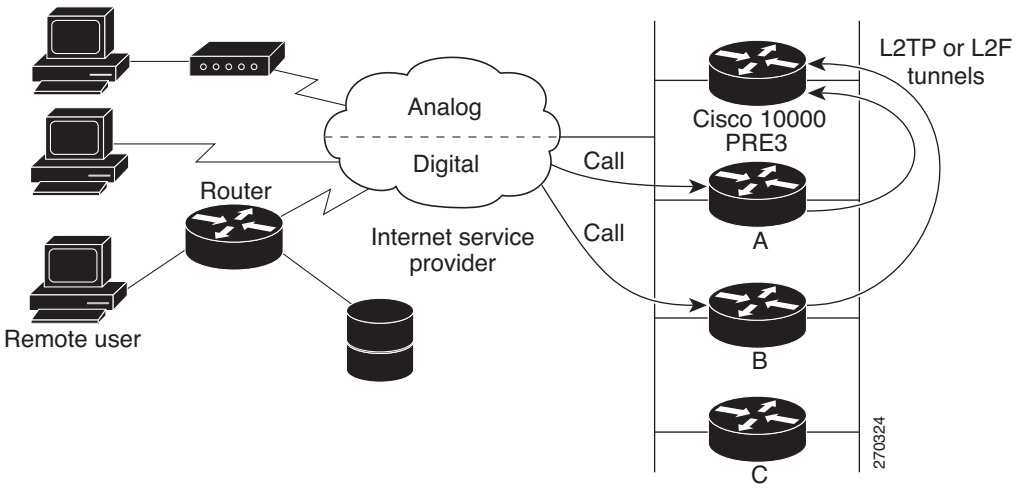
MLP on LNS

Networks are migrating from the Digital Subscriber Line (DSL) aggregation network connectivity to Broadband Remote Access Server (BRAS), with the mix of Ethernet and ATM access network. Therefore, there is an increasing need to support MLP and, Link Fragmentation and Interleaving (LFI), to allow high priority low-latency packets to be interleaved between fragments of lower-priority higher-latency packets. Voice over IP (VoIP) is an example of a low-latency service.

In Cisco 12.2(33) SB release, the MLP on LNS feature has been introduced for Asymmetric Digital Subscriber Line (ADSL) deployments where the upstream bandwidth (BW) is low. The MLP on LNS feature can receive fragments from the customer premises equipment (CPE) ensuring that there is less latency upstream, even if a large packet gets in between the voice packets.

The MLP on LNS feature bundles together a virtual private dial network (VPDN) session on a single logical connection, to form a MLP bundle on the LNS. Prior to Cisco 12.2(33) SB IOS release, C10K supported only the Multilink bundle termination on the PPP Termination Aggregation (PTA) router. In the 12.2(33) SB release, C10K supports MLP termination on the LNS also. However, C10K does not support MLP LAC bundle in this release. Figure 19-3 shows an MLP on LNS application.

Figure 19-3 An MLP on LNS application



The MLP on LNS feature has been described in the following topics:

- About MLP on LNS
- PPP Multilink Link Max Command

About MLP on LNS

The Multilink interface-based configuration requires one virtual-template per bundle, so that the **multilink group #** command can be configured on the virtual-template. However, this is not a feasible configuration for the MLP on LNS feature as you can only scale up to 2000 virtual-templates.

To address the virtual-template scaling issue and to avoid cumbersome configuration management, in the 12.2(33) SB release, Virtual access bundles are supported. In Virtual access bundles, the bundle interface is cloned from the virtual-template when the first member link is negotiated on the LNS. The Virtual access bundle support is limited to bundle termination on LNS.

Prior to the 12.2(33) SB release, Multilink interface-based configuration was used to distinguish between single and multi-member bundles. However, for Virtual access based bundle interface, you can no longer use the interface number range to distinguish between single and multi-member bundles as the bundles are generated dynamically in Cisco IOS 12.2(33)SB release. To distinguish single and multi-member bundles, user specified value for the **ppp multilinks max link #** command is used.

The following two diagrams explain two different MLP on LNS bundle configuration supported with 12.2(33) SB release. Figure 19-4 shows MLP on CPE for Dial-up networks.

Figure 19-4 MLP on LNS - Multi Member Bundle

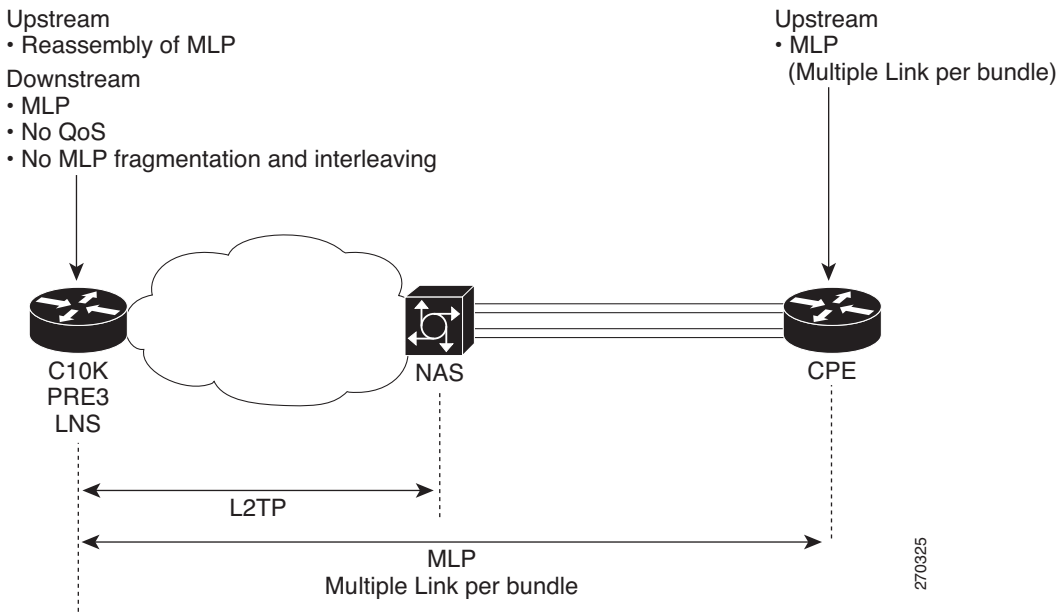
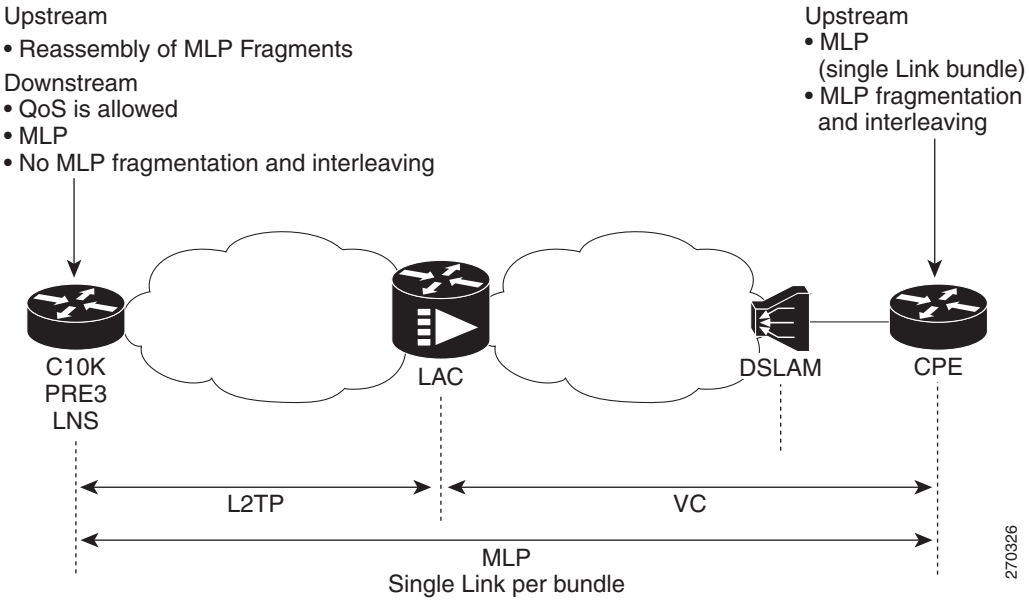


Figure 19-5 shows single member bundle on the CPE. These are single member bundles where the traffic received by the C10k router is fragmented to interleave high priority traffic in between low priority network traffic.

Figure 19-5 MLP on LNS - Single Member bundle



To accommodate the scaling requirements of up to 2040 multi member and 10240 single member bundles for the MLP on LNS feature, an additional reassembly buffer is now reserved in the external column memory (XCM). The reassembly buffer that was reserved in the cobalt space is used for multi-member bundles and XCM reassembly buffer is used for single member bundles.

The fixed reassembly tables size for the MLP on LNS feature to buffer fragments is 256 entries. Reassembly table size restriction has implications on the maximum differential delay for all of the different paths for the member links from CPE to LNS For example, if there are 10 members in a bundle, and one of the members is associated with a "slow" (high delay) path, then the other 9 members must have their fragments/packets buffered while waiting for the slower link. Since the reassembly table stores descriptors, each entry represents one fragment or an whole packet if fragmentation isn't in effect. The amount of time each fragment takes to get transmitted is equal to the configured fragment delay, which is independent of link bandwidth. If fragmentation isn't in effect, the transmit time would be dependent on packet size, with smaller packets being more problematic. Therefore, the amount of differential delay that you can tolerate would be the reassembly table buffering limit for the 9 other links:

(256 / 9) * frag_delay = 28.4 * frag_delay

Note The default differential delay for MLP on LNS is 50ms.

Table 19-4 shows the resource usage on Cisco 10000 series router.

Table 19-4 Resource Usage

	VCCI[1]	HWIDB[2]	SWIDB[3]	PBLT[4]
C10k MAX	64k	Memory dependent	Memory dependent	16k
Bundle interface	1	1	1	1

Member link single member bundle	1	1	1	0
Member link multi-member bundle	1	1	1	1

1. A Virtual Circuit Connection Identifier (VCCI) is a variable that identifies a virtual circuit connection between two nodes.
2. A Hardware Interface Descriptor Block (HWIDB) represents a physical interface, which includes physical ports and channelized interface definitions
3. A Software Interface Descriptor Block (SWIDB) represents a logical sub-interface such as Permanent Virtual Circuit (PVC) or virtual LAN (VLAN)), or a Layer 2 encapsulation (Point-to-Point Protocol (PPP), High-Level Data Link Control (HDLC).
4. A HQF resource that is used by the RP and PXF to program physical layer scheduling for an interface. It could be considered an instance of physical layer scheduling, C10k currently supports 16K such instances. All Bundle interfaces (single or multi-member bundles) use one instance of this resource. For single member bundle the scheduling is done at the logical layer. All members of a multi-member bundles are scheduled at the physical layer, so each member link in a multi-member bundle use one instance.

PPP Multilink Link Max Command

Support for the **ppp multilink link max** command was introduced in 12.2(33)SB, to distinguish between single and multimember MLP on LNS bundle, the default max links for C10k is 10. The **ppp multilink link max 1** command is required for single member bundle. However, the support for this command is limited and has the following restrictions:

- No support for the Frame Relay member link bundle.
- When the command is used on MLP over Serial interface and the **ppp multilink link max 1** command, it restricts the number of links that join the bundle and the bundle continue to use Cobalt for the reassembly space.
- For Single VC and Multi-VC ATM bundles, the command overrides the MLP interface range.

Note The **ppp multilink link max** command is supported only by PRE3.

Performance and Scalability of MLP on LNS

The following commands allow for better scaling when used in configuring MLP on LNS

- Configure the **hold-queue** command in interface configuration mode for the trunk interfaces in which the L2TP tunnel is negotiated, for example:

```
Router(config-if)# hold-queue 4096 in
```

- Configure the following commands and recommended values on the virtual template interface:

```
Router(config-if)# ppp max-configure 110
Router(config-if)# ppp max-failure 100
Router(config-if)# ppp timeout retry 5
Router(config-if)# keepalive 30
```

- Configure the **lcp renegotiation always** command on the VPDN group to renegotiate between LAC and LNS. The maximum multilink member links to be configured on the C10K is up to 20440. Different combinations of bundle configurations can be configured on the box at any given time based on the resource availability.

For more information, see the Scalability and Performance chapter in this guide.

PXF Memory and Performance Impact for MLP on LNS

PXF performance is measured as follows:

- Packet buffer usage

  The number of packet buffers available in the PRE3 is 832K small (for packet sizes of 768 bytes or less) and 120K large (for packet sizes greater than 768 bytes). With full scaling of 12280 bundles (2040 multi and 10240 single link), the average number of buffers is 69.4 small and 10.0 large buffers per bundle for a total of 79.4 buffers per bundle.

  There are 256 entries per bundle. However, in an single link bundle most packets arrive in order, therefore, fewer buffers are required per single link bundle. For example, if there is an average usage of 10 buffers per single link bundle, there are an average of 436.7 buffers per multilink bundle.

- Packet processing rate

  The PRE3 has a rate of 10 million contexts per second which is the rate of contexts or packets passing through the pxf complex. The packet processing rate is measured by the number of packets per second that can either be enqueued or dequeued by the PRE3. If each packet takes 2 passes to be enqueued, then the enqueue rate would be 5 mcps. Since the enqueue and dequeue processing is performed concurrently, the overall performance is determined by the worst case between enqueue and dequeue as shown in Scenario 1 and Scenario 2.

  If the packet processing demand exceeds the available contexts, non-priority packets are dropped.

Scenario 1

- A bi-directional rate of 64kbps per link. Table 19-5 shows the speed performance of a 64kbps link.
- 2040 multilink bundles
- 2, 5 and 10 links per multilink bundle
- 10240 single link bundles
- 200 byte packet size in both directions
- 100 byte fragment size (fragmentation for ingress only)

  Note A frag delay of 2ms requires a 16 byte fragment size. Since the MLP over L2TP header can be on the order of 50 bytes, a 16 byte fragment size is not possible.

Table 19-5 64Kbps link speed performance

Links per multilink bundle	10	5	2
Total links (multi + single)	30640	20440	18400
Total context rate (million contexts/sec)	9.8	6.5	4.6

This scenario shows that for 64Kbps links with a maximum bundle scaling and high demand traffic, the PXF can barely keep up with demand. Therefore, the total number of 64Kbps links should not exceed 20440.

Scenario 2

- A bi-directional rate of 2mbps per link. Table 19-6 shows the speed performance of a 2Mbps link.
- 500 and 2040 multilink bundles
- 2 and 4 links per bundle
- No single link bundles
- 500 and 1000 byte packets in both directions
- 512 byte fragment size (fragmentation for ingress only)

Table 19-6 2Mbps link speed performance (in million contexts per second)

Bundles	2040		500	
Links per bundle	4	2	4	2
Total links	8160	4080	2000	1000
500 byte packets (million context/sec)	24.5	12.2	6.0	3.0
1000 byte packets (million context/sec)	16.3	8.2	4.0	2.0

This scenario shows that for 2Mpbs links with high traffic demand, Cisco 10000 series routers cannot obtain maximum bundle scaling. Therefore, we recommend that the total number of 2Mbps links does not exceed 4080.

Restrictions for MLP on LNS

In Cisco IOS Release 12.2(33)SB, the MLP on LNS feature has the following restrictions:

- No SSO support is planned on the platform for MLP on LNS feature.
- Bundles are only supported with GigE as the trunk between LAC and LNS.
- Bandwidth of the member link has be to received from the LAC through the Connect speed AV-Pair, L2TP sessions on a single link bundle shall be provisioned at the logical layer—HQF. The L2TP sessions on a multi-member MLP bundle shall be provisioned as physical links and shall be bundled at the physical layer (HQF). For multi-member bundle, the bandwidth received through the av-pair is used to carve out the bandwidth from the physical/tunnel interface to reserve it for MLP.
- Oversubscription shall not be supported for MLP bundled L2TP member or on the underlying tunnel interface.
- All member L2TP sessions within the same bundle shall belong to the same physical interface and the same L2TP tunnel
- QoS is supported on a single member MLP bundle. HQoS and Overhead Accounting for MLP bundle is not supported
- QoS on multiple member MLP bundle is not supported. If any MLPoLNS bundles are negotiated on the GigE interface, applying service policy on GigE tunnel interface is also not supported.
- Each member link in a bundle has the same speed. We do not recommend or support configuring member links of different speed.

- Fragmentation and Interleaving on MLP on LNS bundles in the downstream direction are not supported.
- Locally terminated member links and member links forwarded from the LAC are not supported within the same bundle although the setup is not prevented.
- Sessions from different tunnels are not allowed to join the same bundle, all members of a bundle must be part of the same L2TP tunnel and share the same physical interface.
- Multi-Class MLP is not supported for MLP on LNS bundles.
- There shall be a performance impact of one additional toaster phase per fragment added to the MLP on LNS dequeue process.
- Multilink interface based bundles for MLP on LNS are not supported.
- Vaccess bundles support for existing MLP features is not supported in this release.
- Dynamic change of the physical tunnel interface-the GigE on which the L2TP tunnel for MLPoLNS bundle is negotiated due to change in route or switching to backup due to problems on the line is not supported. This will require the bundles to renegotiate.
- For Multi-member bundles, carve out and reserve the bandwidth from the physical interface-the trunk interface on which the L2TP tunnel is negotiated. The bandwidth available for use on the trunk interface or other connection is reduced by the sum of the bandwidth reserved for the bundle.

  Bandwidth available = GigE - BW bundle

  Bandwidth available = GigE - (BW of bundle 1 + BW of bundle 2)
- Packet loss during over subscription and congestion is not deterministic.

Configuring MLP on LNS

You can refer to the following sections for configuring MLP on LNS:

- Required Configuration Tasks for LNS
- Optional Configuration Tasks for LNS

For an configuration example on the MLP on LNS feature, see Configuration Example for MLP on LNS.

MLP-Based Link Fragmentation and Interleaving

MLP supports link fragmentation and interleaving (LFI). The MLP fragmentation mechanism multilink encapsulates large nonreal-time packets and fragments them into a small enough size to satisfy the delay requirements of real-time traffic. Smaller real-time packets are not multilink encapsulated. Instead, the MLP interleaving mechanism provides a special transmit queue (priority queue) for these delay-sensitive packets to allow the packets to be sent earlier than other packet flows. Real-time packets remain intact and MLP interleaving mechanisms send the real-time packets between fragments of the larger non real-time packets.

For more information about link fragmentation and interleaving, see the “Fragmenting and Interleaving Real-Time and Nonreal-Time Packets” chapter in the *Cisco 10000 Series Router Quality of Service Configuration Guide*.

Configuring MLP Bundles and Member Links

Table 19-7 shows the components you must define when configuring MLP (without link fragmentation and interleaving) on specific interface types.

Table 19-7 Requirements for Configuring MLP

Type	MLP Bundle	Member Links	Virtual Template	Service Policy
MLP over Serial	Required	Required	Not Required	Not Required
Single-VC MLP over ATM	Required	Required	Required	Required[1]
Multi-VC MLP over ATM	Required	Required	Required	Required[1]

1. A service policy is required only when configuring MLP-based link fragmentation and interleaving (LFI) for Single-VC or Multi-VC MLP over ATM. For MLP-based LFI, a service policy with a priority queue defined must be attached to the multilink interface. The VC does not require a service policy.

To configure MLP bundles and member links, perform the following configuration tasks:

- Creating an MLP Bundle Interface, page 19-25
- Enabling MLP on a Virtual Template, page 19-27
- Adding a Serial Member Link to an MLP Bundle, page 19-28
- Adding an ATM Member Link to an MLP Bundle, page 19-29

The following configuration tasks are optional:

- Moving a Member Link to a Different MLP Bundle, page 19-32
- Removing a Member Link from an MLP Bundle, page 19-33
- Changing the Default Endpoint Discriminator, page 19-34

Creating an MLP Bundle Interface

To create an MLP bundle interface, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink` *multilink-bundle-number*	Creates a multilink bundle. Enters interface configuration mode to configure the bundle. *multilink-bundle-number* is a nonzero number that identifies the multilink bundle. For Cisco IOS Release 12.2(28)SB and later releases, valid values are: • MLP over Serial—1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). • Single-VC MLP over ATM—10,000 and higher • Multi-VC MLP over ATM—1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). **Note** For releases earlier than to Cisco IOS Release 12.2(28)SB, valid values are from 1 to 2,147,483,647.
Step 2	`Router(config-if)# ip address` *address mask*	Specifies the IP address and subnet mask assigned to the interface. *address* is the IP address. *mask* is the subnet mask for the associated IP address.
Step 3	`Router(config-if)# ppp chap hostname` *hostname*	(Optional) Identifies the hostname sent in the Challenge Handshake Authentication Protocol (CHAP) challenge. *hostname* is the name of the bundle group. This is the unique identifier that identifies the bundle. **Note** If you configure this command on the bundle and its member links, specify the same identifier for both the bundle and the member links.
Step 4	`Router(config-if)# ppp multilink fragment-delay` *delay-max*	(Optional) Configures the maximum delay allowed for the transmission of a packet fragment on an MLP bundle. *delay-max* specifies the maximum amount of time, in milliseconds, that is required to transmit a fragment. Valid values are from 1 to 1000 milliseconds.
Step 5	`Router(config-if)# ppp multilink interleave`	(Optional) Enables interleaving of real-time packets among the fragments of larger nonreal-time packets on an MLP bundle.
Step 6	`Router(config-if)# ppp multilink fragment disable`	(Optional) Disables packet fragmentation.

Note The router automatically adds the **ppp multilink** and **ppp multilink group** commands to the MLP bundle configuration.

Configuration Example for Creating an MLP Bundle Interface

Example 19-1 shows a sample configuration for creating an MLP bundle interface.

Example 19-1 Creating an MLP Bundle Interface

```
Router(config)# interface multilink 8
Router(config-if)# ip address 172.16.48.209 255.255.0.0
Router(config-if)# ppp chap hostname cambridge
```

Enabling MLP on a Virtual Template

The virtual template interface is attached to the member links, not to the MLP bundle. You can apply the same virtual template to the member links; you are not required to apply a unique virtual template to each member link.

To enable MLP on a virtual template, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates or modifies a virtual template interface that can be configured and applied dynamically to virtual access interfaces. Enters interface configuration mode. *number* is a number that identifies the virtual template interface. You can configure up to 5061 total virtual template interfaces (requires Cisco IOS Release 12.2(28)SB and later releases).
Step 2	`Router(config-if)# ppp max-configure` *retries*	Specifies the maximum number of configure requests to attempt before stopping the requests due to no response. *retries* specifies the maximum number of retries. Valid values are from 1 to 255. The default is 10 retries. We recommend 110 retries.
Step 3	`Router(config-if)# ppp max-failure` *retries*	Configures the maximum number of consecutive Configure Negative Acknowledgements (CONFNAKs) to permit before terminating a negotiation. *retries* is the maximum number of retries. Valid values are from 1 to 255. The default is 5 retries. We recommend 100 retries.
Step 4	`Router(config-if)# ppp timeout retry` *response-time*	Sets the maximum time to wait for Point-to-Point Protocol (PPP) negotiation messages. *response-time* specifies the maximum time, in seconds, to wait for a response during PPP negotiation. We recommend 5 seconds.
Step 5	`Router(config-if)# keepalive` [*period*]	Enables keepalive packets to be sent at the specified time interval to keep the interface active. *period* specifies a time interval, in seconds. The default is 10 seconds. We recommend 30 seconds.
Step 6	`Router(config-if)# no ip address`	Removes an IP address.
Step 7	`Router(config-if)# ppp multilink`	Enables MLP on the virtual template interface.

Configuration Example for Enabling MLP on a Virtual Template

Example 19-2 shows a sample configuration for enabling MLP on a virtual template.

Example 19-2 Enabling MLP on a Virtual Template

```
Router(config)# interface virtual-template1
Router(config-if)# ppp max-configure 110
Router(config-if)# ppp max-failure 100
Router(config-if)# ppp timeout retry 5
Router(config-if)# keepalive 30
Router(config-if)# no ip address
Router(config-if)# ip mroute-cache
Router(config-if)# ppp authentication chap
Router(config-if)# ppp multilink
Router(config-if)# exit
```

Adding a Serial Member Link to an MLP Bundle

You can configure up to 10 serial member links per MLP bundle. When adding T1 member links, add only full T1 interfaces. If the interface you add to the MLP bundle contains information such as an IP address, routing protocol, or access control list, the router ignores that information. If you remove the interface from the MLP bundle, that information becomes active again.

To add serial member links to an MLP bundle, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial` *slot/module/port.channel:controller-number*	Specifies the interface that you want to add to the MLP bundle. Enters interface configuration mode. *slot/module/port* identifies the line card. The slashes are required. *channel*: is the channel group number. The colon is required. *controller-number* is the member link controller number.
Step 2	`Router(config-if)# hold-queue` *length* {**in** | **out**}	Limits the size of the IP output queue on an interface. We recommend that you configure this command on all physical interfaces. *length* is a number that specifies the maximum number of packets in the queue. Valid values are from 0 to 4096. We recommend 4096 packets for all line cards. By default, the input queue is 75 packets and the output queue is 40 packets. **in** specifies the input queue. **out** specifies the output queue.
Step 3	`Router(config-if)# ppp max-configure` *retries*	Specifies the maximum number of configure requests to attempt before stopping the requests due to no response. *retries* specifies the maximum number of retries. Valid values are from 1 to 255. The default is 10 retries. We recommend 110 retries.

	Command	Purpose
Step 4	`Router(config-if)# ppp max-failure retries`	Configures the maximum number of consecutive Configure Negative Acknowledgements (CONFNAKs) to permit before terminating a negotiation. *retries* is the maximum number of retries. Valid values are from 1 to 255. The default is 5 retries. We recommend 100 retries.
Step 5	`Router(config-if)# ppp timeout retry response-time`	Sets the maximum time to wait for Point-to-Point Protocol (PPP) negotiation messages. *response-time* specifies the maximum time, in seconds, to wait for a response during PPP negotiation. We recommend 5 seconds.
Step 6	`Router(config-if)# keepalive [period]`	Enables keepalive packets to be sent at the specified time interval to keep the interface active. *period* specifies a time interval, in seconds. The default is 10 seconds. We recommend 30 seconds.
Step 7	`Router(config-if)# ppp chap hostname hostname`	(Optional) Identifies the hostname sent in the Challenge Handshake Authentication Protocol (CHAP) challenge. *hostname* is the name of the bundle group. This is the unique identifier that identifies the bundle. **Note** If you configure this command on the bundle and its member links, specify the same identifier for both the bundle and the member links.
Step 8	`Router(config-if)# encapsulation ppp`	Specifies Point-to-Point Protocol (PPP) encapsulation for the interface.
Step 9	`Router(config-if)# no ip address`	Removes any existing IP address from the main interface.
Step 10	`Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 11	`Router(config-if)# ppp multilink group group-number`	Associates the interface with an MLP bundle. *group-number* is a nonzero number that identifies the multilink group. Valid values are from 1 to 9999. The *group-number* must be identical to the specified *multilink-bundle-number* of the MLP bundle to which you want to add this link.

Adding an ATM Member Link to an MLP Bundle

You can configure up to 10 member links per MLP bundle for Multi-VC MLP over ATM. However, you can configure only one member link per MLP bundle for Single-VC MLP over ATM.

To add ATM member links to an MLP bundle, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm slot/module/port`	Configures or modifies the ATM interface you specify and enters interface configuration mode.
Step 2	`Router(config-if)# hold-queue length {in | out}`	Limits the size of the IP output queue on an interface. We recommend that you configure this command on all physical interfaces, except when using the OC-12 ATM line card. *length* is a number that specifies the maximum number of packets in the queue. Valid values are from 0 to 4096. We recommend 4096 packets for all line cards, except the ATM OC-12 line card. By default, the input queue is 75 packets and the output queue is 40 packets. **in** specifies the input queue. **out** specifies the output queue.
Step 3	`Router(config-if)# interface atm slot/module/port.subinterface point-to-point`	Creates or modifies a point-to-point subinterface. Enters subinterface configuration mode.
Step 4	`Router(config-subif)# ppp chap hostname hostname`	(Optional) Identifies the hostname sent in the Challenge Handshake Authentication Protocol (CHAP) challenge. *hostname* is the name of the bundle group. This is the unique identifier that identifies the bundle. **Note** If you configure this command on the bundle and its member links, specify the same identifier for both the bundle and the member links.
Step 5	`Router(config-subif)# no ip address`	Removes any existing IP address from the main interface.
Step 6	`Router(config-subif)# pvc [name] vpi/vci`	Creates or modifies an ATM PVC. Enters ATM VC configuration mode. *name* is the name of the ATM PVC. vpi/ is the virtual path identifier. If you do not specify a VPI value and the slash character (/), the VPI value defaults to 0. vci is the virtual channel identifier.
Step 7	`Router(config-if-atm-vc)# vbr-nrt output-pcr output-scr output-mbs`	Configures the variable bit rate-nonreal time (VBR-nrt) quality of service (QoS). *output-pcr* is the output peak cell rate (PCR), in kbps. *output-scr* is the sustainable cell rate (SCR), in kbps. *output-mbs* is the output maximum burst cell size (MBS), expressed in number of cells.

	Command	Purpose
Step 8	`Router(config-if-atm-vc)# encapsulation {aal5mux ppp virtual-template number | aal5ciscoppp virtual-template number | aal5snap}`	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM virtual circuit (VC). **aal5mux ppp** specifies the AAL and encapsulation type for multiplex (MUX)-type VCs. The keyword **ppp** is Internet Engineering Task Force (IETF)-compliant PPP over ATM. It specifies the protocol type being used by the MUX encapsulated VC. Use this protocol type for Multi-VC MLP over ATM to identify the virtual template. This protocol is supported on ATM PVCs only. **aal5ciscoppp** specifies the AAL and encapsulation type for Cisco PPP over ATM. Supported on ATM PVCs only. **aal5snap** specifies the AAL and encapsulation type that supports Inverse ARP. Logical Link Control/Subnetwork Access Protocol (LLC/SNAP) precedes the protocol datagram. **virtual-template** *number* is the number used to identify the virtual template.
Step 9	`Router(config-if-atm-vc)# protocol ppp virtual-template number`	Enables PPP sessions to be established over the ATM PVC using the configuration from the virtual template you specify. Use this command only if you specified **aal5snap** as the encapsulation type and you are configuring MLP on multiple VCs. *number* is a nonzero number that identifies the virtual template that you want to apply to this ATM PVC.
Step 10	`Router(config-if-atm-vc)# ppp multilink group group-number`	Associates the PVC with an MLP bundle. *group-number* is a nonzero number that identifies the multilink group. Valid values are: • Single-VC MLP over ATM—10,000 and higher • Multi-VC MLP over ATM—1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). The *group-number* must be identical to the specified *multilink-bundle-number* of the MLP bundle to which you want to add this link.

Configuration Example for Adding ATM Links to an MLP Bundle

Example 19-3 shows how to add ATM links to an MLP bundle. In the example, the virtual template named Virtual-Template 1 is applied to PVCs 0/34, 0/35, and 0/36. Each of these PVCs is assigned to MLP bundle group 1. Notice that all of the member links have the same encapsulation type. The router does not support member links with different encapsulation types.

Example 19-3 Adding ATM Links to an MLP Bundle

```
Router(config)# interface Multilink 1
```

```
Router(config-if)# ip address 10.6.6.1 255.255.255.0
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
!
Router(config)# interface virtual-template1
Router(config-if)# ppp max-configure 110
Router(config-if)# ppp max-failure 100
Router(config-if)# ppp timeout retry 5
Router(config-if)# keepalive 30
Router(config-if)# no ip address
Router(config-if)# ppp multilink
!
Router(config)# interface atm 6/0/0
Router(config-if)# no ip address
Router(config-if)# hold-queue 4096 in
!
Router(config)# interface atm 6/0/0.1 point-to-point
Router(config-if)# no ip address
Router(config-if)# pvc 0/34
Router(config-if-atm-vc)# vbr-nrt 512 256 20
Router(config-if-atm-vc)# encapsulation aal5snap
Router(config-if-atm-vc)# protocol ppp Virtual-Template 1
Router(config-if-atm-vc)# ppp multilink group 1
!
Router(config)# interface atm 6/0/0.2 point-to-point
Router(config-if)# no ip address
Router(config-if)# pvc 0/35
Router(config-if-atm-vc)# vbr-nrt 512 256 20
Router(config-if-atm-vc)# encapsulation aal5snap
Router(config-if-atm-vc)# protocol ppp Virtual-Template 1
Router(config-if-atm-vc)# ppp multilink group 1
!
Router(config)# interface ATM 6/0/0.3 point-to-point
Router(config-if)# no ip address
Router(config-if)# pvc 0/36
Router(config-if-atm-vc)# vbr-nrt 512 256 20
Router(config-if-atm-vc)# encapsulation aal5snap
Router(config-if-atm-vc)# protocol ppp Virtual-Template 1
Router(config-if-atm-vc)# ppp multilink group 1
```

Moving a Member Link to a Different MLP Bundle

To move a member link to a different MLP bundle, enter the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface that you want to move to a different MLP bundle. Enters interface or subinterface configuration mode. *type* specifies the type of interface (for example, ATM). *number* specifies the interface number and is the *slot*/*module*/*port.subinterface* number or the *slot*/*module*/*port.channel*:*controller-number* of the interface (for example, ATM 1/0/0.1).
Step 2	`Router(config-if)# ppp chap hostname hostname`	(Optional) Identifies the hostname sent in the Challenge Handshake Authentication Protocol (CHAP) challenge. *hostname* is the name of the bundle group. This is the unique identifier that identifies the bundle. **Note** If you configure this command on the bundle and its member links, specify the same identifier for both the bundle and the member links.
Step 3	`Router(config-if)# ppp multilink group group-number`	Moves this interface to the MLP bundle you specify. *group-number* identifies the multilink group. Change this *group-number* to the new MLP group *group-number*. Valid values are: • MLP over Serial—1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later). • Single-VC MLP over ATM—10,000 and higher • Multi-VC MLP over ATM—1 to 9999 (Release 12.2(28)SB and later) or from 1 to 9999 and 65,536 to 2,147,483,647 (Release 12.2(31)SB2 and later).

Removing a Member Link from an MLP Bundle

To remove a member link from an MLP bundle, enter the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# interface type number`	Specifies the member link that you want to remove from the MLP bundle. Enters interface configuration mode. *type* specifies the type of interface (for example, ATM). *number* specifies the interface number and is the *slot*/*module*/*port.subinterface* number or the *slot*/*module*/*port.channel*:*controller-number* of the interface (for example, ATM 1/0/0.1).
Step 2	`Router(config-if)# no ppp multilink group group-number`	Removes the member link from the MLP group. *group-number* is the number of the MLP group from which you want to remove the member link.

	Command	Purpose
Step 3	`Router(config-if)# no ppp multilink`	Disables multilink for the link.
Step 4	`Router(config-if)# no ppp chap hostname`	Removes PPP authentication.

Changing the Default Endpoint Discriminator

When the local system negotiates using MLP with the peer system, the default endpoint discriminator value provided is the username that is used for authentication. The **ppp chap hostname** or **ppp pap sent-username** command is used to configure the username for the interface or the username defaults to the globally configured hostname.

To change the default endpoint discriminator, enter the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink endpoint {hostname | ip IP-address | mac LAN-interface | none | phone telephone-number | string char-string}`	Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer system. **hostname** indicates to use the hostname configured for the router. This is useful when multiple routers are using the same username to authenticate, but have different hostnames. **ip** *ip-address* indicates to use the supplied IP address. **mac** *lan-interface* indicates to use the specified LAN interface whose MAC address is to be used. **none** causes negotiation of the Link Control Protocol (LCP) without requesting the endpoint discriminator option, which is useful when the router connects to a malfunctioning peer system that does not handle the endpoint discriminator option properly. **phone** *telephone-number* indicates to use the specified telephone number. Accepts E.164-compliant, full international telephone numbers. **string** *char-string* indicates to use the supplied character string.

Configuration Example for Changing the Endpoint Discriminator

Example 19-4 shows how to change the MLP endpoint discriminator from the default CHAP hostname C-host1 to the hostname cambridge.

Example 19-4 Changing the Default Endpoint Discriminator

```
Router(config)# interface multilink 8
Router(config-if)# ip address 10.1.1.4 255.255.255.0
Router(config-if)# ppp chap hostname C-host1
Router(config-if)# ppp multilink endpoint hostname cambridge
```

Configuration Examples for Configuring MLP

This section provides the following configuration examples:

Configuration Example for Configuring MLP Over Serial Interfaces

Example 19-5 shows a sample configuration for configuring MLP over serial interfaces. In the example, 1/0/0/1:0 and 1/0/0/2:0 subinterfaces are added to the Multilink 1 bundle.

Example 19-5 Configuring MLP on Serial Interfaces

```
interface Multilink1
 ip address 100.1.1.1 255.255.255.0
 no keepalive
 ppp multilink
 ppp multilink group 1
!
interface serial 1/0/0/1:0
 no ip address
 encapsulation ppp
 ppp chap hostname m1
 ppp multilink
 ppp multilink group 1
!
interface serial 1/0/0/2:0
 no ip address
 encapsulation ppp
 ppp chap hostname m1
 ppp multilink
 ppp multilink group 1
```

Configuration Example for Configuring Multi-VC MLP Over ATM

Example 19-6 shows a sample configuration for configuring Multi-VC MLP over ATM. In the example, PVC 0/36 on ATM subinterface 5/0/0.3 and PVC 0/37 on ATM subinterface 5/0/0.4 are added to the Multilink 2 bundle. The virtual template named Virtual-Template1 is applied to PVC 0/36 and PVC 0/37.

Example 19-6 Configuring Multi-VC MLP Over ATM VCs

```
interface Multilink2
 ip address 100.1.2.1 255.255.255.0
 ppp multilink
 ppp multilink group 2
!
interface ATM5/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM5/0/0.3 point-to-point
 pvc 0/36
  ppp chap hostname m2
  ppp multilink group 2
  vbr-nrt 128 64 20
```

```
  encapsulation aal5mux ppp Virtual-Template1
!
interface ATM5/0/0.4 point-to-point
 pvc 0/37
  ppp chap hostname m2
  ppp multilink group 2
  vbr-nrt 128 64 20
  encapsulation aal5mux ppp Virtual-Template1
!
interface Virtual-Template1
 no ip address
 no keepalive
 ppp max-configure 110
 ppp max-failure 100
 ppp multilink
 ppp timeout retry 5
!
```

Configuration Example for MLP on LNS

Example 19-7 shows how to setup a tunnel on the GigabitEthernet interface on which the VPDN member links are negotiated and added to the MLP bundle cloned from virtual template 500.

Example 19-7 MLP on LNS

```
aaa new-model
!
!
aaa authentication ppp default local
aaa authentication ppp TESTME group radius
aaa authorization network default local
aaa authorization network TESTME group radius
!
aaa session-id common

buffers small perm 15000
buffers mid perm 12000
buffers big perm 8000


!
vpdn enable
!
vpdn-group LNS_1
 accept-dialin
  protocol l2tp
  virtual-template 500
 terminate-from hostname LAC1-1
 local name LNS1-1
 lcp renegotiation always
 l2tp tunnel receive-window 100
 L2tp tunnel password 0 cisco
 l2tp tunnel nosession-timeout 30
 l2tp tunnel retransmit retries 7
 l2tp tunnel retransmit timeout min 2
 l2tp tunnel retransmit timeout max 8
!
!
interface GigabitEthernet2/0/0
 ip address 210.1.1.3 255.255.255.0
```

```
 negotiation auto
 hold-queue 4096 in
!
!
interface Virtual-Template500
 ip unnumbered Loopback1
 peer default ip address pool pool-1
 ppp mtu adaptive
 ppp timeout authentication 100
 ppp max-configure 110
 ppp max-failure 100
 ppp timeout retry 5
keepalive 30
 ppp authentication pap TESTME
 ppp authorization TESTME
 ppp multilink
!
ip local pool pool-1 1.1.1.1 1.1.1.100

radius-server host 15.1.0.100 auth-port 1645 acct-port 1646 key cisco
radius-server retransmit 0
```

Verifying and Monitoring MLP Connections

To verify and monitor MLP connections, enter the following commands in privileged EXEC mode:

Command	Purpose
Router# **debug ppp multilink events**	Displays information about events affecting multilink groups established for Bandwidth Allocation Control Protocol (BACP).
Router# **show atm pvc**	Displays all ATM permanent virtual circuits (PVCs) and traffic information.
Router# **show interfaces** *type number*	Displays statistics for the interface you specify. If you do not specify a specific interface, statistics display for all interfaces configured on the router.
Router# **show interfaces virtual-access** *number* [**configuration**]	Displays status, traffic data, and configuration information about the virtual access interface you specify. **Note** This command currently displays statistics for system traffic only. Statistics for bundle traffic do not display. For information about bundle traffic, see the **show interfaces** or **show ppp multilink** command. *number* is the number of the virtual access interface. (Optional) **configuration** restricts output to configuration information.

Command	Purpose
`Router# show interfaces multilink group-number [stat]`	Displays configuration information about the MLP bundle you specify. *group-number* is a nonzero number that identifies the multilink bundle. (Optional) **stat** indicates to display traffic statistics for the MLP bundle such as the number of packets in and out.
`Router# show ppp multilink [bundle-interface]`	Displays bundle information for all of the MLP bundles and their PPP links configured on the router. (Optional) *bundle-interface* specifies the multilink interface (for example, Multilink 5). If you specify *bundle-interface*, the command displays information for only that specific bundle.
`Router# show running-config`	Displays information about the current router configuration, including information about each interface configuration.

Bundle Counters and Link Counters

When you enter the **show interface** command on an MLP bundle interface and on all of its member link interfaces, you might expect that the counters on the bundle should be equal to the sum of the counters for all of the link interfaces. However, this is not the case.

The statistics for the various interfaces reflect the data that actually goes through those interfaces. The data that goes through the bundle is different than the data going through the links. All of the traffic at the bundle-level does eventually pass through the link-level, but it is not formatted the same. In addition, links also carry traffic that is private to that link, such as link-level keepalives.

The following describes some of the reasons link-level and bundle-level counts might be different (ignoring the link-private traffic):

- Multilink fragmentation might be occurring. A single packet at the bundle level becomes multiple packets at the link level.
- Frames at the bundle level include only bundle-level encapsulation, which consists of a 2-byte PPP header (or 1-byte under some circumstances).
- Frames at the link-level include link-level encapsulation bytes, which includes all forms of media-specific encapsulation and framing. This includes headers and trailers for High Level Data Link Control (HDLC) and PPP over ATM. The link-level encapsulation bytes also include multilink subheaders (for example, sequence numbers), if they are used.

Note Multilink subheaders are not part of the packet encapsulation as it exists at the bundle level. Multilink subheaders are part of the encapsulation that is added to fragments before placing them on the link; they are not added to the network-level datagrams (for example, IP packets) prior to sending them to the fragmentation engine.

Because of the factors listed above, the counts on the links can be greater than the counts on the bundle. The link level has a great deal of overhead that is not visible at the bundle level.

Verification Examples for MLP Connections

This section provides the following verification examples:

Verification Example for the show interfaces multilink Command

Example 19-8 shows sample output for the **show interfaces multilink** command. In the example, configuration information and packet statistics display for the MLP bundle 8.

Example 19-8 Sample Output for the show interfaces multilink Command

```
Router# show interfaces multilink 8
Multilink8 is up, line protocol is up
Hardware is multilink group interface
Internet address is 10.1.1.1/24
MTU 1500 bytes, BW 15360 Kbit, DLY 100000 usec, rely 255/255, load
1/255
Encapsulation PPP, crc 16, loopback not set
Keepalive not set
DTR is pulsed for 2 seconds on reset
LCP Open, multilink Open
Open:IPCP
Last input 15:24:43, output never, output hang never
Last clearing of "show interface" counters 15:27:59
Queueing strategy:fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
36 packets input, 665 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
31 packets output, 774 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

Verification Example for the show ppp multilink Command

Example 19-9 shows sample output from the **show ppp multilink** command. In the example, information about the MLP over ATM bundle (Multilink3) displays first. Information about the member links then displays, including the number of active and inactive member links. Class fields are omitted from the output; everything is implicitly in receive class 0 and transmit class 0.

Example 19-9 Sample Output for the show ppp multilink Command

```
Router# show ppp multilink

Multilink3, bundle name is multilink_name-3
  Endpoint discriminator is multilink_name-3
  Bundle up for 3d21h, total bandwidth 128, load 1/255
  Receive buffer limit 24384 bytes, frag timeout 1000 ms
  Bundle is Distributed
    0/0 fragments/bytes in reassembly list
```

```
    1 lost fragments, 1 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x831D received sequence, 0x0 sent sequence
  C10K Multilink PPP info
    Bundle transmit info
        send_seq_num   0x0
    Bundle reassembly info
        expected_seq_num:  0x00831E
  Member links: 2 active, 0 inactive (max 10, min not set)
    Vi5, since 3d21h, 16 weight, 82 frag size
    Vi4, since 3d19h, 16 weight, 82 frag size No inactive multilink interfaces
```

The following describes the bundle-level fields and lines in the **show ppp multilink** command output:

- Bundle name is *name*—The bundle identifier for the bundle.
- Bundle up for *time*—The elapsed time since the bundle first came up.
- load *n*/255—The traffic load on the bundle as multilink computes loads for bandwidth-on-demand purposes. This load might count all traffic, or just inbound or outbound traffic, depending on the configuration.
- Receive buffer limit *n* bytes—The maximum amount of fragment data that multilink can buffer in its fragment reassembly engine for each receive class. This is derived from the configured slippage constraints.
- Frag timeout *n* ms—The maximum amount of time that multilink waits for an expected fragment before declaring it lost. This only applies when fragment loss cannot be detected by other, faster means such as sequence number-based detection.
- Member links:—The number of active and inactive links currently in the bundle, followed by the desired minimum and maximum number of links. The actual number might be outside the range.

After all of the bundle parameters display, information about each individual link in the bundle displays. Extra link level parameters might be shown after each link in certain circumstances. The following describes the individual link parameters:

- Weight—The weight is used for load balancing purposes. Data is distributed between the member links in proportion to their weight. The weight is proportional to multilink's notion of the effective bandwidth of a link. Therefore, multilink effectively distributes data to the links in proportion to their bandwidth.

  Multilink's notion of the effective bandwidth of a link is the configured bandwidth value, except on asynchronous lines where multilink uses a value that is 0.8 times the configured bandwidth setting. This is because on an asynchronous line, at best only 8-tenths (8/10) of the raw bandwidth is available for transmitting real data and the remainder is consumed in framing overhead.

  Previously, the weight also controlled the size of the fragments generated for that link. However, Cisco IOS software now computes a separate fragment size value.
- Frag size—The size of the largest fragment that can be generated for that link. It is the size of the Multilink PPP (MLP) payload carried by a fragment and does not include MLP headers or link level framing.
- Unsequenced—This indicates that the serial link is unsequenced and packets can arrive in a different order than the peer transmitted them. To compensate for this, multilink relaxes its lost fragment detection mechanisms.
- Receive only (or receive only pending)—This indicates that the link is in idle mode or is about to be put in idle mode. Processing of arriving data on the link continues normally, but data is not transmitted on the link. The remote system is expected to not send data on the link.

Verification Example for the show interfaces multilink stat Command

Example 19-10 shows sample output for the **show interfaces multilink stat** command. In the example, the number of packets in and out display for each of the specified switching paths.

Example 19-10 Sample Output for the show interfaces multilink stat Command

```
Router# show interfaces multilink 8 stat
Multilink 8
Switching pathPkts InChars InPkts OutChars Out
    Processor3666531774
    Route cache000 0
    Total36 665 31  774
```

Related Documentation

This section provides hyperlinks to additional Cisco documentation for the features discussed in this chapter. To display the documentation, click the document title or a section of the document highlighted in blue. When appropriate, paths to applicable sections are listed below the documentation title.

Feature	Documentation
Multilink PPP	Cisco IOS Dial Services Configuration Guide: Terminal Services, Release 12.1 Part 4: PPP Configuration > Configuring Media-Independent PPP and Multilink PPP
MLP over ATM	RFC 1990, *The PPP Multilink Protocol* *Designing and Deploying Multilink PPP over Frame Relay and ATM* tech notes
MLP over Serial	RFC 1990, *The PPP Multilink Protocol*

Feature	Documentation
Link Fragmentation and Interleaving	Cisco 10000 Series Router Quality of Service Configuration Guide Fragmenting and Interleaving Real-Time and Nonreal-Time Packets Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits, Release 12.1(5)T feature module Cisco IOS Quality of Service Solutions Configuration Guide Link Efficiency Mechanisms > Link Efficiency Mechanisms Overview > Link Fragmentation and Interleaving for Frame Relay and ATM VCs Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits RFC 1990, *The PPP Multilink Protocol*
PPP Encapsulation	RFC 1661, The Point-to-Point Protocol Cisco IOS Wide-Area Networking Configuration Guide, Release 12.2 Configuring Broadband Access: PPP and Routed Bridge Encapsulation

CHAPTER 20

Configuring Gigabit EtherChannel Features

On a Cisco 10000 Series router, a Gigabit EtherChannel (GEC) is a specialized interface type comprising aggregated Gigabit Ethernet links. A GEC bundle is synonymous with port channel and can have a minimum of one or a maximum of 8 active links. The bandwidth of the GEC interface is the aggregate of all the physical member links comprising the GEC bundle.

Note Cisco IOS Release 12.2(31)SB supports a maximum of 4 member links per GEC bundle. In Cisco IOS Release12.2(15)BX, the maximum number of links per GEC bundle has been increased from 4 to 8.

The Gigabit EtherChannel can be deployed in two ways on the Cisco 10000 Series router:

- *Core facing* or network facing deployment is an uplink EtherChannel that connects the Cisco 10000 Series router to the service provider. This setup has multiple physicals links bundled per GEC interface and allows:
  - Load balancing across all the active interfaces.
  - Combination of different Gigabit Ethernet (GE) ports (both shared port adaptors and line cards)
- *Access facing* or subscriber facing deployment connects the Cisco 10000 Series router to the subscriber edge. This setup typically has only one active member link on a GEC bundle interface. The remaining links in the GEC bundle serve as passive links. Traffic is sent only through the active member link, while the passive link is used as a backup when the active member link fails. This arrangement provides link redundancy with no loss of Point-to-Point Protocol over Ethernet (PPPoE) sessions during link failover.
  - Load balancing is not applicable when there is only one active link in a GEC bundle.
  - Queuing action is allowed only on the GEC bundle and not on member links.

Note A GEC bundle can include a combination of active and passive links. In an M:N mode, ‘M’ denotes active links and ‘N’ denotes passive links. In a 1:N mode only one link is active per GEC bundle and ‘N’ denotes passive links. Passive links operate as backup links but do not transfer any network traffic.

This chapter describes Gigabit EtherChannel (GEC) enhancements implemented on the Cisco 10000 Series routers and includes the following topics:

For more information on Gigabit EtherChannels (GEC) see, the *Link Aggregation Control Protocol (LACP) (802.3ad) for Gigabit Interfaces* feature guide at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/10gigeth.htm

Feature History for Gigabit EtherChannel Enhancements

Cisco IOS Release	Description	Required PRE
12.2(31)SB	This feature was supported only on native line cards. • QoS policies supported only on GEC member links.	PRE2 and PRE3
12.2(33)SB	This feature is supported on native line cards[1] and the SPA Interface Processor (SIP) and Shared Port Adapters (SPA)[2] on the Cisco 10000 Series router.	PRE3 and PRE4
12.2(33)SB	The following Gigabit EtherChannel enhancements were added on the Cisco 10000 Series router: • QoS Service Policies on GEC Bundle[3] • PPPoE hitless[4] switchover support with Link Aggregation Control Protocol (LACP) (802.3ad) port channel • PBR Support for GEC Bundle • IEEE 802.1Q and QinQ Support on GEC Bundle • Multicast VPN Support on GEC Bundle • PPPoX Support for GEC Bundle (includes PPPoEoE, PPPoEoQinQ, PPPoVLAN) • High availability for SSO, NSF, ISSU, and NSR • 8 Member Links per GEC bundle	PRE2, PRE3, and PRE4

1. 1 Port GE – Half-height and 1 Port GE – Full-height.
2. 2 Port GE – Half-height and 5 Port GE – Half-height.
3. Queuing actions still require QoS application on each member link.
4. A hitless switchover implies that no PPPoX sessions are dropped when an active member link fails and the passive or backup link takes over as the active link.

Prerequisites for Gigabit EtherChannel Configuration

The following are the prerequisites for configuring GEC bundles:

- Create a GEC bundle interface before adding GE links to the GEC bundle using the **channel-group** command.
- Add GE links to the GEC bundle and configure all the links identically.

Restrictions for Gigabit EtherChannel Configuration

The following are general restrictions applicable to GEC bundles:

- Bidirectional Forwarding Detection Protocol (BFD) is not supported on GEC bundles.
- A dot1Q/QinQ subinterface created on a GEC bundle requires one VCCI on all the member links. The GEC bundle also uses one VCCI.
- Intelligent Service Gateway (ISG) IP sessions are not supported on GEC bundles.
- Gigabit EtherChannels are not supported on a 10 GE shared port adaptor (SPA).
- On core facing deployments, queuing actions are only supported when they are applied on each individual member link.
- In 1:N deployment mode:
  - Combination of any SPA based Gigabit Ethernet ports with either a full height GE line card or half height GE line card is not supported.
  - A maximum of 4 Gigabit Ethernet SPAs form a GEC bundle. Each SPA interface must have the same bay number and port number, assuming the representation is GigabitEthernetSlotNumber/BayNumber/Port-Number.

    For example, in the case of SPAs, Gi1/2/1 can be bundled with Gi5/2/1, but Gi1/2/1 cannot be bundled with Gi1/0/1.
  - Member link counters are not updated.
- We do not recommend the deletion of the GEC bundle main interface before removing the member links, using the **no channel-group** command. Similarly, before you delete the GEC bundle main interface, remove the subinterfaces.

Configuring QoS Service Policies on GEC Interfaces

The QoS support feature on Gigabit EtherChannel allows service policies to be applied on GEC traffic. Support for QoS can be applied differently to GEC bundles or main interfaces of member links and GEC bundle subinterfaces.

- Input QoS can directly be applied on GEC bundle subinterfaces or main interfaces. Input QoS can be applied on member links using the vlan-group QoS feature. For details, see the "Configuration Example for Using the VLAN Group Feature to Apply QoS on Member Links" section on page 20-5 and the "Configuration Example for Applying QoS on GEC Bundle Subinterfaces" section on page 20-6.
- Output QoS can directly be applied on GEC bundle subinterfaces similar to the GEC main interfaces. Alternatively, output QoS can be applied on member links using the vlan-group QoS feature. The service policy with **match-vlan class-maps** is applied on the member link main interface.

The application of QoS depends on the deployment mode of the GEC bundle interface as described in the Table 20-1:

Table 20-1 Service Policies Applied on GEC Bundles

QoS for GEC	M:N Deployment	1:N Deployment
Input QoS for GEC	Service policy can be applied for both GEC bundle interface and member links. If applied on a: • GEC bundle, the aggregate ingress traffic on GEC bundle is subject to this service-policy. • Member link, ingress traffic on a particular member link is subject to the service-policy applied on the member link.	Service policy can be applied only on the GEC bundle interface.
Output QoS for GEC	Service-policy *without* queuing actions can be applied either on GEC bundle interface or on the member links. If applied on a: • GEC bundle, the aggregate egress traffic on GEC bundle is subject to this service-policy. • Member link, egress traffic on a particular member link is subject to the service-policy applied on the member link. Service-policies *with* queuing actions can be applied only on member links.	Service policies *with* or *without* queuing actions can be applied only on the GEC bundle interface.
Input QoS for GEC subinterface	Input QoS applied on GEC bundle interface and on member main-interfaces. If the service-policy is applied on a: • GEC bundle subinterface, the aggregate ingress traffic on the GEC bundle subinterface is subject to this service-policy. • GEC member main-interface using vlan-group feature, the ingress traffic on that member link with the vlan-ids specified in the vlan-group service-policy is subject to the corresponding actions as specified in the service-policy.	Input QoS can only be applied on a bundle subinterface. The aggregate bundle subinterface traffic is subject to this service-policy.
Output QoS for GEC subinterface	Service-policies *without* queuing actions can be applied either on the GEC bundle subinterface or on the member main-interface.If the service-policy is applied on a: • GEC bundle subinterface, the aggregate egress traffic on that GEC bundle subinterface is subject to this service-policy. • GEC member main-interface using vlan-group feature, the egress traffic on that member link with the vlan-ids specified in the vlan-group service-policy is subject to the corresponding actions as specified in the service-policy. Service policies *with* queuing actions can only be applied on member links. The egress traffic on that member link with the vlan-ids specified in the vlan-group service-policy is subject to the corresponding actions as specified in the service-policy.	Output QoS *with* or *without* queuing actions can only be applied on a GEC bundle subinterface.

Restrictions for QoS Service Policies on GEC Bundles

The following restrictions are applicable to QoS service policies applied on GEC bundle interfaces and subinterfaces:

- Both ingress and egress service-policy without any queuing actions can only be applied on member links for M:N deployment, and are restricted for 1:N deployment.
- Egress service-policy with queuing action can only be applied on:
  - Member interfaces for an M:N GEC deployment
  - Bundle interface for a 1:N GEC deployment.
- Restriction for application of QoS on VLAN groups:
  - A VLAN group restricts the application of QoS to a maximum of 255 individual dot1Q (VLAN) subinterfaces. 255 denotes the number of user classes that can be used in VLAN group parent policy.
  - Matching a VLAN group user class on QinQ subinterfaces is not supported using VLAN group policy. Each VLAN group user class at parent hierarchy can only match a set of dot1Q subinterfaces.
  - VLAN group QoS policy at member link is used only when dot1Q subinterfaces are defined on the GEC bundles and not when QinQ subinterfaces are defined on the GEC bundle.
- Input Quality of Service (QoS) on member links is not supported for QinQ subinterfaces.
- The classification criteria of **match input-interface port-channel** is not supported. Instead, packets are classified by matching them with member-links.

Configuration Examples

This section provides the following configuration examples:

- Configuration Example for Using the VLAN Group Feature to Apply QoS on Member Links, page 20-5
- Configuration Example for Applying QoS on GEC Bundle Subinterfaces, page 20-6

Configuration Example for Using the VLAN Group Feature to Apply QoS on Member Links

Step 1 Consider a GEC bundle interface with two member links Gig3/0/0 and Gig4/0/0.

Step 2 Assume subinterfaces exist on the GEC bundle interface having the following configurations:

```
Port-channel 1.1
    Encapsulation dot1q 2
Port-channel 1.2
    Encapsulation dot1q 3
Port-channel 1.3
```

Encapsulation dot1q 4

Assume that the following configurations need to be performed on each member link

- Police ingress traffic for subinterface port-channel 1.1 at 100 mbps
- Police ingress traffic for subinterface port-channel 1.2 at 150 mbps
- Shape egress traffic for subinterface port-channel 1.2 at 50 mbps
- Shape egress traffic for subinterfaces port-channel 1.1 and port-channel 1.3 together at 150 mbps

Step 3 Create **match-vlan class-maps** as follows:

```
Class-map match-any vlan_2
    Match vlan 2
```

```
Class-map match-any vlan_3
    Match vlan 3
Class-map match-any vlan_4
    Match vlan 4
Class-map match-any vlan_2_4
    Match vlan 2 4
```

Step 4 Create **policy-maps** as follows:

```
Policy-map mega_ingress
    Class vlan_2
        Police 100 mbps
    Class vlan_3
        Police 150 mbps
Policy-map mega_egress
    Class vlan_3
        Shape 50 mpbs
    Class vlan_2_4
        Shape 150 mbps
```

Step 5 Apply this policy on the GEC member links.

```
Interface Gig3/0/0
    Service-policy input mega_ingress
    Service-policy output mega_egress
Interface Gig4/0/0
    Service-policy input mega_ingress
    Service-policy output mega_egress
```

Configuration Example for Applying QoS on GEC Bundle Subinterfaces

Example 20-1 shows how QoS is applied on GEC bundle subinterfaces:

Example 20-1 Applying QoS on GEC Bundle Subinterfaces

```
Class-map match-any dscp_20_30
    Match dscp 20 30
Class-map match-any dscp_40
    Match dscp 40

Policy-map police_dscp
   Class dscp_20_30
          Police 50 3000 3000 conform-action transmit exceed-action drop
    Set ip dscp af22
   Class dscp_40
         Police 10 3000 3000 conform-action transmit exceed-action drop

Policy-map customer_A
  Class class-default
    Police 100 mpbs
    service-policy police_dscp

Policy-map customer_B
  Class class-default
    Police 150 mbps
    Service-policy police_dscp

Interface Port-channel 1.1
    Service-policy input customer_A

Interface Port-channel 1.2
```

```
Service-policy input customer_B
```

Configuring Policy Based Routing Support on a GEC Bundle

Cisco Policy Based Routing (PBR) provides a flexible mechanism for network administrators to customize the operation of the routing table and the flow of traffic within their networks.

Load balancing is performed on packets that pass through PBR. If a PBR clause is applied to outbound packets, and the clause results in the selection of an EtherChannel egress interface, then a new hash is generated based on the new address information. This hash is used to select an appropriate member link as the packet's final destination.

Policy based routing is supported on Gigabit EtherChannel. You can configure PBR directly on a GEC bundle interface.

Restriction for Configuring PBR Support on a GEC Bundle

- Use of the **set interface** command is restricted in the set clause for PBR, on GEC bundle interfaces. Only the IP address for the next hop can be specified.

Configuring IEEE 802.1Q and QinQ Support on GEC Bundle

Support for both dot1Q and QinQ subinterfaces is available for GEC bundle interfaces. Configuring subinterface on a GEC bundle interface is similar to a normal Gigabit Ethernet interface configuration. When a subinterface is configured on a GEC bundle interface, the GEC *Toaster* client creates subinterface instances in the Toaster for all the GEC member links. The subinterface instances created internally on GEC member links are hidden and not available to the user for applying configurations.

Note The *toaster* is designed to process IP packets at very high rates using existing forwarding algorithms, though it may also be programmed to perform other tasks and protocols.

Prerequisites for Configuring IEEE 802.1Q and QinQ Support

- Create a GEC bundle main interface before creating subinterfaces.

Restrictions for Configuring IEEE 802.1Q and QinQ Support on GEC Bundle

- A dot1Q/QinQ subinterface created on GEC bundle requires a VCCI on all the member-links.

  For example, a GEC bundle interface with 8 member-links uses 9 (1+8) VCCIs for each dot1Q/QinQ subinterface created on the GEC bundle.
- Ingress packet accounting for QinQ subinterfaces is carried out at the bundle level. Accounting of these ingress packets per member link is not supported.

Configuration Tasks for IEEE 802.1Q and QinQ on Subinterfaces

To create a GEC bundle subinterface and configure a dot1Q/QinQ encapsulation, enter the following commands beginning from the global configuration mode in the following table:

	Command	Purpose
Step 1	`router(config)# interface port-channel number`	Creates a GEC bundle.
Step 2	`router(config)# interface port-channel subinterface`	Creates a GEC bundle subinterface and enters the subinterface mode.
Step 3	`router(config-subif)# encapsulation dot1Q vlan-id`	Enables IEEE 802.1Q encapsulation on a specified subinterface in a VLAN.
Step 4	`router(config-subif)# encapsulation dot1Q vlan-id second-dot1q inner vlan-id`	Enables 802.1Q encapsulation on a specified subinterface in an inner VLAN.
Step 5	`router# show running-config interface port-channel subinterface`	Displays the current configuration for the GEC bundle subinterface.
Step 6	`router# show interface port-channel subinterface`	Displays status, traffic data, and configuration information about the subinterface you specify.

Configuration Examples

Example 20-2 and Example 20-3 show the encapsulation configuration details:

Example 20-2 show interface Command for the GEC Bundle Subinterface

```
router# show interface port-channel 1.1
Port-channel1.1 is up, line protocol is up
  Hardware is GEChannel, address is 0004.9b3e.101a (bia 0004.9b3e.1000)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  20.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
```

Example 20-3 show running-config Command for the GEC Bundle Subinterface

```
router# show running-config interface port-channel 1.1
Building configuration...

Current configuration : 134 bytes
!
interface Port-channel1.1
 encapsulation dot1Q 20 second-dot1q 200
 ip address 3.0.0.1 255.255.255.0
 end
```

Configuring MVPN Support on GEC Bundle

The Multicast VPN (MVPN) feature allows a service provider to configure and support multicast traffic within a Virtual Private Network (VPN) environment. MVPN also supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

On the Cisco 10000 Series router, when we use GEC as a core facing link (from the provider edge to the provider) the MVPN packet sent on the GEC interface has the IP header encapsulated inside a GRE Header or the Tunnel Header. The hash function is calculated based on the tunnel header's source and destination IP address and the original IP header's (Inner IP header) source and destination address. Load balancing is used on outbound GEC packets to find out the member link on which the packet is sent.

For more information on MVPN, see the "IP Multicast VPN" section in the *Multicast VPN—IP Multicast Support for MPLS VPN* guide at:

http://www.cisco.com/en/US/docs/ios/12_2sb/feature/guide/sbb_mvpn.html#wp1040907

Configuration Tasks and Examples

For configuration information and examples, see the "How to Configure Multicast VPN—IP Multicast Support for MPLS VPNs" section in the *How to Configure Multicast VPN—IP Multicast Support for MPLS VPNs* at:

http://www.cisco.com/en/US/docs/ios/12_2sb/feature/guide/sbb_mvpn.html#wp1041284

Configuring PPPoX Support on a GEC Bundle

PPPoE, PPPoEoQinQ, PPPoEoVLAN sessions are supported only in 1:N GEC mode and are provisioned on the GEC bundle interface. The complete session traffic is directed towards the active member link. Hence, when the active link goes down, the session traffic is directed towards the passive member link, which then becomes the active link.

PPPoX sessions on GEC work similar to that of a normal Gigabit Ethernet interface and QoS policy inheritance is similar to that of a normal Gigabit Ethernet interface.

Restrictions for Configuring PPPoX Support for GEC Bundle

- Support for PPPoX sessions is allowed only in a 1:N mode, where there is only one active GEC link.
- At any point of time the bandwidth of the 1:N GEC bundle will be 1 Gbps.

Note Multiple passive links can be added, but only one active link is supported for PPPoE.

For more information on PPPoEoQinQ support for subinterfaces, see PPPoE - QinQ Support feature guide at:

http://www.cisco.com/en/US/docs/ios/12_3t/12_3t7/feature/guide/gt_qinq.html

Configuration Tasks

To enable PPPoE session creation on a GEC bundle, enter the following commands:

	Command	Purpose
Step 1	`router(config)# interface port-channel number`	Creates a GEC bundle.
Step 2	`router(config)# lacp max-bundle 1-8`	Sets the maximum number of active links per GEC bundle. For PPPoE sessions maximum number of active links is one.
Step 3	`router(config)# lacp fast-switchover`	Retains PPPoX sessions incase of member link switchover.
Step 4	`router(config)# interface port-channel subinterface`	Creates a GEC bundle subinterface and enters the subinterface mode.
Step 5	`router(config-subif)# encapsulation dot1Q vlan-id`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN. Specify the VLAN identifier.
Step 6	`router(config-subif)# pppoe enable group global`	Enables PPPoE session on the GEC bundle subinterface. **global** is the default group used when a group name is not specified.
Step 7	`router(config-subif)# end`	Exits to the global configuration mode.

For more information on PPPoE over Ethernet, see the *Cisco 10000 Series Router Software Configuration Guide* at:

http://www.cisco.com/en/US/docs/routers/10000/10008/configuration/guides/broadband/vlan.html

Configuration Examples

Example 20-4 shows how to enable a PPPoE session on a GEC bundle:

Example 20-4 Enabling a PPPoE Session

```
interface Port-channel32
no ip address
no negotiation auto
lacp max-bundle 1
lacp fast-switchover
!
interface Port-channel32.1
encapsulation dot1Q 10
pppoe enable group bba_group_1
!
interface Port-channel32.2
encapsulation dot1Q 20
pppoe enable group bba_group_1
!
```

Configuring High Availability Support on GEC Bundle

The following high availability features are supported on GEC bundle interfaces, on the Cisco 10000 Series router.

- Stateful Switchover (SSO)
- In Service Software Upgrade (ISSU))
- Nonstop Forwarding (NSF)
- Nonstop Routing (NSR)

The EtherChannel and the IEEE 802.3ad LACP protocol are SSO and ISSU aware. This feature makes the GEC bundle interface available after a PRE switchover, in the event of a catastrophic failure.

For more information on NSF, see *Cisco Nonstop Forwarding* whitepaper at:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsnsf20s.html

For information on SSO see, *Stateful Switchover* feature guide at:

http://www.cisco.com/en/us/docs/ios/12_2s/feature/guide/fssso20s.html

For more information on ISSU, see *Cisco IOS In Service Software Upgrade and Enhanced Fast Software Upgrade Process* feature guide, at:

http://www.cisco.com/en/US/docs/ios/12_2sb/feature/guide/sb_issu.html

Configuring 8 Member Links per GEC Bundle

A maximum of 8 configured member links are supported per GEC bundle on the Cisco 10000 Series router. The number of member links per GEC bundle has been increased from 4 to 8 in the Cisco IOS Release 12.2(15)BX.

Configuration Tasks

The following table lists the configuration commands used to configure the maximum and minimum links on a GEC bundle on a Cisco 10000 Series router.

Command	Purpose
`router(config-if)# lacp max-bundle 1-8`	Assigns the maximum number of links per bundle. The default value is 8.
`router(config-if)# lacp min-link 1-8`	Sets the minimum number of active links required before declaring the port channel interface down. The default value is 1.
`router(config-if)# lacp fast-switchover`	Reduces the switchover time from 2 seconds to 10ms. By default lacp fast-switchover configuration is not enabled.

For more information on configuring member links, see the *Link Aggregation Control Protocol (LACP) (802.3ad) for Gigabit Interfaces* feature guide at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/10gigeth.htm

For more information on how to aggregate multiple Ethernet links into one logical channel, see *IEEE 802.3ad Link Bundling* feature guide at:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/sbcelacp.htm#wp1053782

CHAPTER 21

Configuring IP Version 6

Internet Protocol version 6 (IPv6), formerly called IPng (next generation), is the latest version of IP. IPv6 offers many advantages over the previous version of IP, including a larger address space. IPv6 has been available on other Cisco platforms; with the release of Cisco IOS release 12.2(28)SB, it is available on the Cisco 10000 series routers running the PRE2 processor.

For information about how to configure and use these IPv6 features on Cisco platforms, see the *Cisco IOS IPv6 Implementation Library*, located at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00805766e4.html.

This chapter the following information for the IPv6 feature:

Feature History for IPv6

Cisco IOS Release	Description	Required PRE
12.2(28)SB	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(31)SB2	Support was added for the PRE3. Support was added for extended ACLs.	PRE3

Supported Features

The Cisco 10000 series routers support the following IPv6 PXF features:

- Coexistence with IPv4
- IPv6 Addressing
- IPv6 extension header. PXF handling of extension headers includes:
  - Diversion of packets with hop-by-hop extension header

  - Ability to match on fragment and presence of routing headers
  - Skipping extension headers to get to layer 4 information
  - Flag setting to match on the “undetermined-transport” ACL flag
- IPv6 Internet Control Message Protocol (ICMP)
- IPv6 NDP
- IPv6 Layer 2 encapsulation:
  - Point to Point Protocol (PPP)
  - Multilink PPP
  - High-level Data Link Control (HDLC)
  - VLAN
  - Point-to-point Frame Relay
  - Point-to-point ATM
- IPv6 Routing:
  - Static
  - Routing Information Protocol (RIPng)
  - Open Shortest Path First (OSPFv3)
  - Border Gateway Protocol (BGP4+)
  - Intermediate System-to-Intermediate System (ISISv6)
- IPv6 Tunneling (manual and generic routing encapsulation (GRE))
  - Manually configured bi-directional IPv6-in-IPv4 GRE tunnels
  - Manually configured bi-directional IPv4-over-IPv4 tunnels
    Maximum of 1000 IPIP or GRE tunnels
- HA/ISSU coexistence; IPv6 support is RPR+
- IPv6 Unicast Forwarding

  The Cisco 10000 series router maintains the following global (unless otherwise specified) IPv6-specific packet counters:
  - forwarded—number of IPv6 packets forwarded
  - no adjacency—number of IPv6 packets punted due to adj_index=0. Statistics per VCCI will be collected for this specific punt case (diversion cause).
  - adj_discard— number of IPv6 packets dropped due to discard adjacency. Statistics per VCCI will be collected for this specific drop (column 5).
  - adj_punt—number of IPv6 packets punted due to punt adjacency
  - adj_glean— number of IPv6 packets punted due to glean adjacency
  - adj_drop—number of IPv6 packets punted due to drop adjacency (RP generates ICMP and drops after that)
  - adj_null—number of IPv6 packets punted due to null adjacency
  - adj_receive—number of IPv6 packets punted due to receive adjacency
  - adj_unknown—number of IPv6 packets punted due to unknown adjacency (e.g. 0x80)
- Strict Reverse Path Forwarding (RPF)

RPF strict check mode verifies that the source IP address exists in the FIB table and verifies that the source IP address is reachable through the input port

- Security ACLs

  For IPv6, ACEs include the following new fields:

  - Flow Label
  - Presence of Routing Header
  - "Undetermined Transport"

- QoS

  QoS matching is performed only on the following subset of fields, which are common to IPv4 and IPv6:

  - dscp/precedence
  - access group (matches only on ACE entries common to IPv4 and IPv6)
  - class
  - qos group
  - mpls
  - input if
  - l2 cos
  - discard class

  The **match protocol** command now includes the **ipv6** keyword to specify this protocol as a matching criterion. The **match ip dscp** and **match ip precedence** commands apply only to IPv4 traffic. The **match dscp** and **match precedence** commands apply to both IPv4 and IPv6 traffic.

  For marking packets, the **set ip dscp** and **set ip precedence** commands have been changed to **set dscp** and **set precedence**. They now apply to both IPv4 and IPv6 traffic.

- ICMP handling and generation are performed on the route processor and are not handled in PXF

Limitations for IPv6

Not all types of IPv6 Tunneling are supported on the Cisco 10000 routers with this release. Among those not supported are the following:

- Automatic 6to4
- ISATAP
- Automatic IPv4-compatible
- IPv6 over L2TPv3
- 6over4 (RFC 2529)
- IPv6 in IPv6 GRE
- IPv6 over UTI

The following security ACL features are not supported for IPv6:

- Incremental compilation (The Cisco 10000 routers use pre-compiled ACLs.)
- Single-step classification

- ACL logging
- Time-based ACLs
- Reflexive ACLs
- Receive Path ACLs
- MiniACLs

QoS matching is not provided on the following two fields, which are IPv6-specific:

- IPv6 src/dst address
- IPv6 ACL

IPv6 Extended ACLs

Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode.

In Cisco IOS Release 12.2(31)SB2 and later releases, the standard IPv6 ACL functionality is extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

Prerequisites

In Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases, for backward compatibility, the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode is still supported; however, an IPv6 ACL defined with deny and permit conditions in global configuration mode is translated to IPv6 access list configuration mode. See the "Create and Apply IPv6 ACL: Examples" section for an example of a translated IPv6 ACL configuration.

Restrictions

IPv6 ACLs are defined by a unique name (IPv6 does not support numbered ACLs). An IPv4 ACL and an IPv6 ACL cannot share the same name.

- Each IPv6 ACL contains implicit permit rules to enable IPv6 neighbor discovery. These rules can be overridden by the user by placing a **deny ipv6 any any** statement within an ACL. The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, makes use of a separate data link layer protocol; therefore, by default, IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

Configuring IPv6 Traffic Filtering

To enable IPv6 traffic filtering, you must perform the following steps:

1. Create an IPv6 ACL
2. Configure the IPv6 ACL to pass or block traffic
3. Apply the IPv6 ACL to an interface

Creating and Configuring the IPv6 ACL

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
   or
   deny *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Router(config)# ipv6 access-list outbound`	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#. • The *access-list name* argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.

Step 4	**permit** *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
	
	or
	
	deny *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]
	
	Example:
	`Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 eq telnet any reflect reflectout`
	
	Example:
	`Router(config-ipv6-acl)# deny tcp host 2001:0db8:1::1 any log-input`
		Specifies permit or deny conditions for an IPv6 ACL.
		• The *protocol* argument specifies the name or number of an Internet protocol. It can be one of the keywords **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **sctp**, **tcp**, or **udp**, or an integer in the range from 0 to 255 representing an IPv6 protocol number.
		• The *source-ipv6-prefix*/*prefix-length* and *destination-ipv6-prefix*/*prefix-length* arguments specify the source and destination IPv6 network or class of networks about which to set permit conditions.
		Note These arguments must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
		• The **any** keyword is an abbreviation for the IPv6 prefix ::/0.
		• The **host** *source-ipv6-address* keyword and argument specify the source IPv6 host address about which to set permit conditions.
		• The *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
		For information on supported arguments and keywords, see the **permit** and **deny** commands in the *IPv6 for Cisco IOS Command Reference* document.

Applying the IPv6 ACL to an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 traffic-filter** *access-list-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic-filter** *access-list-name* {**in** | **out**} **Example:** `Router(config-if)# ipv6 traffic-filter outbound out`	Applies the specified IPv6 access list to the interface specified in the previous step. • The **in** keyword filters incoming IPv6 traffic on the specified interface. • The **out** keyword filters outgoing IPv6 traffic on the specified interface.

Verifying IPv6 ACLs

In the following example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Router> show ipv6 access-list

IPv6 access list inbound
    permit tcp any any eq bgp reflect tcptraffic (8 matches) sequence 10
    permit tcp any any eq telnet reflect tcptraffic (15 matches) sequence 20
    permit udp any any reflect udptraffic sequence 30

IPv6 access list tcptraffic (reflexive) (per-user)
    permit tcp host 2001:0DB8:1::32 eq bgp host 2001:0DB8:2::32 eq 11000 timeout 300 (time
        left 243) sequence 1
    permit tcp host 2001:0DB8:1::32 eq telnet host 2001:0DB8:2::32 eq 11001 timeout 300
        (time left 296) sequence 2

IPv6 access list outbound
    evaluate udptraffic
    evaluate tcptraffic
```

Note For a description of each output display field, see the **show ipv6 access-list** command in the *IPv6 for Cisco IOS Command Reference* document.

Create and Apply IPv6 ACL: Examples

The following example configures two IPv6 ACLs named OUTBOUND and INBOUND and applies both ACLs to outbound and inbound traffic on Ethernet interface 0. The first and second permit entries in the OUTBOUND list permit all TCP and User Datagram Protocol (UDP) packets from network 2001:0DB8:0300:0201::/32 to exit out of Ethernet interface 0. The entries also configure the temporary IPv6 reflexive ACL named REFLECTOUT to filter returning (incoming) TCP and UDP packets on Ethernet interface 0. The first deny entry in the OUTBOUND list keeps all packets from the network fec0:0:0:0201::/64 (packets that have the site-local prefix fec0:0:0:0201 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0.

The **evaluate** command in the INBOUND list applies the temporary IPv6 reflexive ACL named REFLECTOUT to inbound TCP and UDP packets on Ethernet interface 0. When outgoing TCP or UDP packets are permitted on Ethernet interface 0 by the OUTBOUND list, the INBOUND list uses the REFLECTOUT list to match (evaluate) the returning (incoming) TCP and UDP packets.

```
ipv6 access-list OUTBOUND
 permit tcp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 permit udp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 deny fec0:0:0:0201::/64 any

ipv6 access-list INBOUND
 evaluate REFLECTOUT

interface ethernet 0
 ipv6 traffic-filter OUTBOUND out
 ipv6 traffic-filter INBOUND in
```

Note Given that a **permit any any** statement is not included as the last entry in the OUTBOUND or INBOUND ACL, only TCP and UDP packets matching the configured permit entries in the ACL and ICMP packets matching the implicit permit conditions in the ACL are permitted out of and in to Ethernet interface 0 (the implicit deny all condition at the end of the ACL denies all other packet types on the interface).

The following example configures HTTP access to be restricted to certain hours during the day, and to log any activity outside of the permitted hours.

```
time-range lunchtime
 periodic weekdays 12:00 to 13:00

ipv6 access-list OUTBOUND
 permit tcp any any eq www time-range lunchtime
 deny tcp any any eq www log-input
 permit tcp 2001:0DB8::/32 any
 permit udp 2001:0DB8::/32 any
```

CHAPTER 22

Configuring Template ACLs

When user profiles are configured using RADIUS Attribute 242, similar per-user access control lists (ACLs) may be replaced by a single Template ACL. That is, one ACL represents many similar ACLs. In Cisco IOS Release 12.2(28)SB, by using Template ACLs, you can increase the total number of ACLs used in the Cisco 10000 series routers but minimize the memory and CPU consumption in processing the ACLs.

The Template ACL feature is useful for customers in a broadband environment with tens of thousands of subscribers. Network implementations that use a unique ACL for each subscriber can easily exceed the maximum available resources on the Cisco 10000 series routers. In networks where each subscriber has its own ACL, it is common for the ACL to be the same for each user except for the user's IP address. Template ACLs alleviate this problem by grouping ACLs with many common access control elements (ACEs) into a single ACL that compiles faster and saves system resources. By using the Template ACL feature, service providers can provision unique ACLs for up to 60,000 subscribers using RADIUS Attribute 242. Configuration of ACLs remains the same as in previous Cisco IOS versions.

For example, the following example shows two ACLs that can be sent using Attribute 242, for two separate users:

```
ip access-list extended Virtual-Access1.1#1
permit igmp any host 1.1.1.1
permit icmp host 1.1.1.1 any
deny ip host 44.33.66.36 host 1.1.1.1
deny tcp host 1.1.1.1 44.33.66.36
permit udp any host 1.1.1.1
permit udp host 1.1.1.1 any
permit udp any host 192.168.2.1
permit udp any host 192.170.2.1
permit icmp host 42.55.15.4 host 192.168.2.1
permit udp 11.22.11.0 0.0.0.255 host 192.177.2.1
permit tcp any host 192.170.2.1
permit ip host 42.55.15.4 host 192.168.2.1
permit tcp 11.22.11.0 0.0.0.255 host 192.177.2.1


ip access-list extended Virtual-Access1.1#2
permit igmp any host 13.1.1.2
permit icmp host 13.1.1.2 any
deny ip host 44.33.66.36 host 13.1.1.2
deny tcp host 13.1.1.2 44.33.66.36
permit udp any host 13.1.1.2
permit udp host 13.1.1.2 any
permit udp any host 192.168.2.1
permit udp any host 192.170.2.1
permit icmp host 42.55.15.4 host 192.168.2.1
permit udp 11.22.11.0 0.0.0.255 host 192.177.2.1
permit tcp any host 192.170.2.1
```

```
permit ip host 42.55.15.4 host 192.168.2.1
permit tcp 11.22.11.0 0.0.0.255 host 192.177.2.1
```

With the Template ACL feature enabled, these two ACLs can be recognized as similar, and a new Template ACL is created as follows:

```
ip access-list extended 4_Temp_<random-number>
permit igmp any host <PeerIP>
permit icmp host <PeerIP> any
deny ip host 44.33.66.36 host <PeerIP>
deny tcp host <PeerIP> 44.33.66.36
permit udp any host <PeerIP>
permit udp host <PeerIP> any
permit udp any host 192.168.2.1
permit udp any host 192.170.2.1
permit icmp host 42.55.15.4 host 192.168.2.1
permit udp 11.22.11.0 0.0.0.255 host 192.177.2.1
permit tcp any host 192.170.2.1
permit ip host 42.55.15.4 host 192.168.2.1
permit tcp 11.22.11.0 0.0.0.255 host 192.177.2.1
```

In this example, therefore, an IP address would be associated as follows:

- Virtual-Access1.1#1 1.1.1.1
- Virtual-Access1.1#2 13.1.1.2

The PXF engine knows which user a packet is coming from or going to, so it can get the user IP for comparison from the IP address table.

Template ACLs are activated only for per-user ACLs configured through RADIUS Attribute 242. Any other ACL type is not subject to Template ACL processing. The Template ACL feature is enabled by default, and all Attribute 242 ACLs are considered for template status.

Using the **access-list template** *number* command, you can limit Template ACL status to only ACLs with *number* or fewer rules. The default setting is 100 rules; this value is larger than most Attribute 242 ACLs.

The Template ACLs feature is described in the following topics:

Feature History for Template ACLs

Cisco IOS Release	Description	Required PRE
12.2(28)SB	This feature was introduced on the Cisco 10000 series router.	PRE2
12.2(31)SB2	Supported was added for the PRE3.	PRE3

Configuration Tasks for Template ACLs

If ACLs are configured using RADIUS Attribute 242, Template ACLs are enabled by default. Configuration tasks for Template ACLs include the following:

- Configuring the Maximum Size of Template ACLs (Optional)
- Configuring ACLs Using RADIUS Attribute 242

Configuring the Maximum Size of Template ACLs (Optional)

By default, Template ACL status is limited to ACLs with 100 or fewer rules. You can set this number lower.

To configure the maximum number of rules in Template ACLs, enter the following command in global configuration mode:

```
Router(config)# access-list template number
```

The range for *number* is from 1 to 100.

Example 22-1 shows the configuration of Template ACL processing for individual user ACLs with 50 or fewer rules.

Example 22-1 Configuring a Template ACL

```
Router(config)# access-list template 50
Router(config)#
```

Configuring ACLs Using RADIUS Attribute 242

Template ACL processing occurs only for ACLs that are configured using RADIUS Attribute 242. Attribute 242 has the following format for an IP data filter:

Ascend-Data-Filter = "ip <dir> <action> [dstip <dest_ipaddr\subnet_mask>] [srcp <src_ipaddr\subnet_mask>] [<proto> [dstport <cmp> <value>] [srcport <cmp> <value>] [<est>]]"

Table 22-1 describes the elements in an Attribute 242 entry for an IP data filter.

Table 22-1 IP Data Filter Syntax Elements

Element	Description
ip	Specifies an IP filter.
<dir>	Specifies the filter direction. Possible values are **in** (filtering packets coming into the router) or **out** (filtering packets going out of the router).
action	Specifies the action the router should take with a packet that matches the filter. Possible values are **forward** or **drop**.

Table 22-1 IP Data Filter Syntax Elements (continued)

Element	Description
dstip <dest_ipaddr\subnet_mask>	Enables destination-IP-address filtering. Applies to packets whose destination address matches the value of **<dest_ipaddr>**. If a subnet mask portion of the address is present, the router compares only the masked bits. If you set **<dest_ipaddr>** to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
srcp<src_ipaddr\subnet_mask>	Enables source-IP-address filtering. Applies to packets whose source address matches the value of **<src_ipaddr>**. If a subnet mask portion of the address is present, the router compares only the masked bits. If you set **<src_ipaddr>** to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
<proto>	Specifies a protocol specified as a name or a number. Applies to packets whose protocol field matches this value. Possible names and numbers are **icmp** (**1**), **tcp** (**6**), **udp** (**17**), and **ospf** (**89**). If you set this value to zero (0), the filter matches any protocol.
dstport <cmp> <value>	Enables destination-port filtering. This keyword is valid only when **<proto>** is set to **tcp** (**6**) or **udp** (**17**). If you do not specify a destination port, the filter matches any port. **<cmp>** defines how to compare the specified **<value>** to the actual destination port. This value can be **<**, **=**, **>**, or **!**. **<value>** can be a name or a number. Possible names and numbers are **ftp-data (20)**, **ftp (21)**, **telnet (23)**, **nameserver (42)**, **domain (53)**, **tftp (69)**, **gopher (70)**, **finger (79)**, **www (80)**, **kerberos (88)**, **hostname (101)**, **nntp (119)**, **ntp (123)**, **exec (512)**, **login (513)**, **cmd (514)**, and **talk (517)**.
srcportcmp <cmp> <value>	Enables source-port filtering. This keyword is valid only when **<proto>** is set to **tcp** (**6**) or **udp** (**17**). If you do not specify a source port, the filter matches any port. **<cmp>** defines how to compare the specified **<value>** to the actual destination port. This value can be **<**, **=**, **>**, or **!**. **<value>** can be a name or a number. Possible names and numbers are **ftp-data** (**20**), **ftp** (**21**), **telnet** (**23**), **nameserver** (**42**), **domain** (**53**), **tftp** (**69**), **gopher** (**70**), **finger** (**79**), **www** (**80**), **kerberos** (**88**), **hostname** (**101**), **nntp** (**119**), **ntp** (**123**), **exec** (**512**), **login** (**513**), **cmd** (**514**), and **talk** (**517**).
<est>	When set to 1, specifies that the filter matches a packet only if a TCP session is already established. This argument is valid only when **<proto>** is set to **tcp (6)**.

Example 22-2 shows four Attribute 242 IP data filter entries.

Example 22-2 RADIUS Attribute 242 IP Data Filter Entries

```
Ascend-Data-Filter="ip in drop"
Ascend-Data-Filter="ip out forward tcp"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16
dstport!=telnet"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16 icmp"
```

Monitoring and Maintaining the Template ACL Configuration

To monitor and maintain the configuration of the Template ACL feature, enter any of the following commands in EXEC mode:

Command	Purpose
`Router# show access-list template summary`	Displays information about all Template ACLs.
`Router# show access-list template` *`acl-name`*	Displays information about the named Template ACL.
`Router# show access-list template exceed` *`number`*	Displays the name of all Template ACLs serving as the parent for more than *number* child ACLs.
`Router# show access-list template tree`	Displays information about the entries in the Red-Black data tree.
`Router# show pxf cpu access security`	Displays PXF security ACL statistics. This command does not display statistics for individual child ACLs that are associated with a Template ACL. This command displays the Template ACL parent, with the total statistics for all the associated children ACLs.

Configuration Examples for Template ACLs

Template ACLs are activated only for per-user ACLs configured through RADIUS Attribute 242. For more examples of configuring RADIUS attributes, see Chapter 16, "Configuring RADIUS Features."

access-list template Command

To enable Template ACL processing, use the **access-list template** command in global configuration mode. To disable Template ACL processing, use the **no** form of the command.

The Template ACL feature is enabled by default. The default number of rules for Template ACL status is 100, which is larger than most ACLs configured using Attribute 242.

Command	Purpose
`Router(config)# access-list template number`	Enables Template ACL processing. *number* specifies the maximum length of ACL that should be considered for template status. Only ACLs with *number* or fewer rules will be considered for template status. If the *number* variable is omitted, the default of 100 will be used, and only ACLs with 100 or fewer rules will be considered for template status. Default is 100 rules.

access-list template Command History

Cisco IOS Release	Description
12.2(28)SB	This command was introduced on the Cisco 10000 series router.

access-list template Command Modes

Use this command in global configuration mode.

Usage Guidelines for the access-list template Command

Reducing the number of rules for Template ACL status can lower CPU utilization. The process of checking each ACL against other known ACLs in the system is easier if the matching task can be aborted earlier. However, if you set the number too low (smaller than the largest "similar" Attribute 242 ACL), CPU utilization can go very high, because ACLs that previously would be considered as Template ACL duplicates are now sent to the PXF without regard to other ACLs already in the router.

Setting the number of rules higher can increase CPU utilization, because the comparison task takes some CPU.

Note Changes in CPU utilization occur only during session initiation. Steady-state CPU utilization is unaffected by these changes in ACL processing.

Examples

The following example specifies that ACLs with more than 50 rules will be considered for Template ACL status:

```
Router# access-list template 50
```

show access-list template Command

To display information about Template ACLs, use the **show access-list template** command in EXEC mode.

Command	Purpose
`Router# show access-list template {summary | aclname | exceed number | tree}`	Displays information about ACLs. **summary** displays summary information. *aclname* displays information about the specified ACL. **exceed** *number* identifies Template ACLs that replace more than *number* individual ACLs. **tree** provides an easily readable summary of the frequency of use of each of the ACL types that the Template ACL function sees Output from this command includes the following information for each entry on the Red-Black tree: • CRC32 value • For each ACL associated with a particular CRC32: – Primary ACL name – Number of users of that ACL

show access-list template Command Modes

Use the **show access-list template** command in EXEC mode.

show access-list template Command History

Cisco IOS Release	Description
12.2(28)SB	This command was introduced on the Cisco 10000 series router.

Examples

This section provides examples of the different forms of the **show access-list template** command.

show access-list template summary

The following example shows output from the **show access-list template summary** command:

```
Router# show access-list template summary
Maximum rules per template ACL = 100
Templates active = 1
Number of ACLs those templates represent = 50
Number of tree elements = 1
```

Output from this command includes:

- Maximum number of rules per Template ACL
- Number of discovered active templates
- Number of ACLs replaced by those templates

show access-list template *aclname*

The following examples show output from the **show access-list template** *aclname* command.

```
Router# show access-list template 4Temp_1073741891108

    Showing data for 4Temp_1073741891108
    4Temp_1073741891108 peer_ip used is 172.17.2.62,
    is a parent, attached acl count = 98
    currentCRC = 59DAB725


Router# show access-list template 4Temp_1342177340101

    Showing data for 4Temp_1342177340101
    4Temp_1342177340101 idb's ip peer = 172.17.2.55,
    parent is 4Temp_1073741891108, user account attached to parent = 98
    currentCRC = 59DAB725
```

Output from this display includes:

- Peer IP of the interface associated with the named Template ACL
- Name of the ACL serving as the primary user of the named Template ACL
- Number of ACLs matching the template of the named Template ACL
- Current cyclic redundancy check 32-bit (CRC32) value

show access-list template exceed *number*

The following example shows output from the **show access-list template exceed** *number* command:

```
Router# show access-list template exceed 49
ACL name                          OrigCRC    Count     CalcCRC
4Temp_#120795960097               104FB543   50      104FB543
```

Table 2 describes the significant fields shown in the display.

Table 2 show access-list template exceed Field Descriptions

Field	Description
ACL Name	Name of the ACL that serves as the primary ACL for each template that exceeds *number* ACLs
OrigCRC	Original CRC32 value
Count	Count of ACLs that match the Template ACL
CalcCRC	Calculated CRC32 value

show access-list template tree

The following example shows output from the **show access-list template tree** command:

```
Router# show access-list template tree

ACL name               OrigCRC    Count   CalcCRC
4Temp_1073741891108    59DAB725   98      59DAB725
```

Table 3 describes the significant fields shown in the display.

Table 3 show access-list template tree Field Descriptions

Field	Description
ACL name	Name of an ACL on the Red-Black tree
OrigCRC	Original CRC32 value
Count	Number of users of the ACL
CalcCRC	Calculated CRC32 value

CHAPTER 23

Protecting the Router from DoS Attacks

Internet service providers (ISPs) and other Cisco customers face increasing Denial of Service (DoS) attacks associated with IP options set in the IP header of packets. Cisco IOS routers use the Route Processor (RP) to process IP options packets, which can become problematic during a DoS attack. To protect the router, the Cisco 10000 series router supports the dropping of packets with IP options.

This chapter discusses the following topics:

IP Options Selective Drop

The IP Options Selective Drop feature enables you to protect your network routers in the event of a denial of service (DoS) attack. Hackers who initiate such attacks commonly send large streams of packets with IP options. By dropping the packets with IP options, you can reduce the load of IP options packets on the router. The end result is a reduction in the effects of the DoS attack on the router and on downstream routers.

Internet service providers (ISPs) and other Cisco customers face increasing DoS attacks associated with IP options set in the IP header. Cisco IOS routers are susceptible to DoS attacks because of the way in which the routers process IP options. The hardware-based forwarding engine of Cisco IOS routers cannot handle IP options; therefore, the forwarding engine forwards the IP options packets to the route processor (RP). Similarly, most of the line cards forward IP option packets to the RP. The software-based RP processes the packets and performs the extra processing that the IP options packets require.

Processing IP options packets in the RP can become problematic. Software-switching of IP options packets can lead to a serious security problem if a Cisco IOS router comes under a DoS attack by a hacker sending large streams of packets with IP options. The RP can easily become overloaded and drop high priority or routing protocol packets. Switching packets in software slows down the switching speed of the router and increases the router's vulnerability to resource saturation. Some types of IP options, such as the Router Alert option, can be especially harmful to the router when forwarded to the RP.

By default, Cisco IOS software processes packets with IP options, as required by RFC 1812, *Requirements for IP Version 4 Routers*. The IP Options Selective Drop feature provides the ability to drop packets with IP options in the forwarding engine so that they are not forwarded to the RP. This result in a minimized load on the RP and reduced RP processing requirements.

Feature History for IP Options Selective Drop

Cisco IOS Release	Description
12.0(23)S	This feature was introduced.
12.2(2)T	This feature was integrated in Cisco IOS Release 12.2(2)T.
12.2(25)S	This feature was integrated in Cisco IOS Release 12.2(25)S.
12.2(27)SBC	This feature was integrated in Cisco IOS Release 12.2(27)SBC.
12.3(19)	This feature was integrated in Cisco IOS Release 12.3(19).
12.2(31)SB2	This feature was integrated in Cisco IOS Release 12.2(31)SB2 and introduced on the Cisco 10000 series router for the PRE2 and PRE3.

Restrictions for IP Options Selective Drop

Resource Reservation Protocol (RSVP), Multiprotocol Label Switching-Traffic Engineering (MPLS-TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP options packets may not function in drop mode if this feature is configured.

How to Configure IP Options Selective Drop

You can configure the router to drop all the inbound IPv4 packets with IP options or all the RP-forwarded IP options packets.

To configure IP Options Selective Drop and protect the RP during a DoS attack, perform the following configuration tasks:

- Dropping Packets with IP Options, page 23-2
- Verifying IP Options Packets, page 23-3

Dropping Packets with IP Options

Use the following procedure to configure the forwarding engine to drop packets with IP options before sending them to the RP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip options drop**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip options drop** **Example:** `Router(config)# ip options drop`	Turns IP options processing off. The router drops all the packets received with IP options. **Note** To resume normal options processing, use the **no** form of the command: **no ip options**.

Verifying IP Options Packets

Use the **show ip traffic** command to verify that the router drops all the packets received with IP options.

Configuration Examples for IP Options Selective Drop

This section provides the following configuration examples:

- Dropping IP Options Packets: Example, page 23-3
- Verifying IP Options Handling: Example, page 23-4

Dropping IP Options Packets: Example

The following sample configuration shows how to configure the router (and downstream routers) to drop all the packets with IP options that enter the network:

```
Router(config)# ip options drop

% Warning:RSVP and other protocols that use IP Options packets may not function in drop or
ignore modes.
end
```

Verifying IP Options Handling: Example

The following sample output from the **show ip traffic** command indicates that the router received 2905 packets with IP options set. Because the **ip options drop** command is configured, the router drops all the packets with IP options, as indicated by the options denied counter.

```
Router# show ip traffic

IP statistics:
  Rcvd:  2905 total, 13 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 1 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 12 received, 3 sent
  Mcast: 0 received, 0 sent
  Sent:  3 generated, 0 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop, 0 unsupported-addr
         3000 options denied, 0 source IP address zero
```

Related Documentation

This section provides additional Cisco documentation for the features discussed in this chapter. To display the documentation, click the document title or a section of the document highlighted in blue. When appropriate, paths to applicable sections are listed below the documentation title.

Feature	Related Documentation
Denial of service (DoS) attacks	*Characterizing and Tracing Packet Floods Using Cisco Routers technical note*

CHAPTER 24

IP Tunneling

This chapter describes IP tunneling features implemented on the Cisco 10000 series routers and includes the following topics:

GRE Tunnel IP Source and Destination VRF Membership

The Generic Routing Encapsulation (GRE) Tunnel IP Source and Destination VRF Membership feature enables both unicast and multicast traffic from subscribers to traverse a tunnel interface on the router and terminate in a VRF. Instead of the termination point being in the global routing table, the tunnel's source and destination endpoints terminate within a non-global VRF.

The following software enhancements provide the functionality required to implement this feature:

For more information, see the *GRE Tunnel IP Source and Destination VRF Membership, Release 12.2(31)SB5* feature guide, located at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

Tunnel VRF

The Tunnel VRF feature allows you to terminate GRE tunnels in a virtual private network (VPN) routing and forwarding (VRF) instance. Using this feature, you can configure the source and destination of a tunnel to belong to any VRF table.

A VRF table stores routing data for each VPN. The VRF table defines the VPN membership of a customer site attached to the network access server (NAS). Each VRF table comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

Previously, GRE IP tunnels required the IP tunnel destination to be in the global routing table. Tunnel VRF enables you to configure the tunnel source and destination to belong to any VRF. Like existing GRE tunnels, the tunnel is disabled if no route to the tunnel destination is defined.

The **tunnel vrf** command is used to configure the Tunnel VRF feature. The VRF specified in the **tunnel vrf** command is the same VRF as the VRF associated with the physical interface over which the tunnel sends packets. This provides outer IP packet routing.

For more detailed information, see the *Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership* feature module, located at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

VRF-Aware VPDN Tunnels

The VRF-Aware VPDN Tunnels feature allows you to create VPDN tunnels that use a customer VRF address from a VRF routing table as the tunnel endpoint. The VPDN tunnel terminates in one VRF and the associated PPP sessions terminate in a different VRF. For example, VPDN tunnels can start outside the Multiprotocol Label Switching (MPLS) VPN and terminate within the MPLS VPN.

To configure VRF-Aware VPDN Tunnels, use the **vpn** command in VPDN configuration mode. This command specifies that the source and destination IP addresses of a given VPDN group belong to the specified VRF. Before you enter the **vpn** command, you must first create the VRF instance using the **ip vrf** command. Different VRF-aware VPDN tunnels can have overlapping IP addresses across VRF instances.

The **ip vrf forwarding** command, configured in tunnel interface mode, enables VRF forwarding on an interface. The VRF associated with the tunnel in the **ip vrf forwarding** command configuration is the VRF that the packets are to be forwarded in as the packets exit the tunnel. This provides inner IP packet routing.

The Cisco 10000 series router supports the VRF-Aware VPDN Tunnels feature on the PRE2 and PRE3and applies to the router when acting as the L2TP access concentrator (LAC) or a Layer 2 network server (LNS). As a LAC or an LNS, the router can initiate and terminate tunnels within a specified VRF.

For more information, see the *VRF-Aware VPDN Tunnels* feature module, located at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

Feature History for GRE Tunnel IP Source and Destination VRF Membership

Cisco IOS Release	Description	Required PRE
12.3(7)XI7	This feature was integrated into Cisco IOS Release 12.3(7)XI7, enhanced to include domain support, and implemented on the LAC.	PRE2
12.2(28)SB	This feature was integrated in Cisco IOS Release 12.2(28)SB.	PRE2
12.2(31)SB5	This feature was introduced on the PRE3.	PRE3
12.2(33)SB	Support for VRF-Aware VPDN tunnels feature on LNS	PRE3 and PRE4

Restrictions for GRE Tunnel IP Source and Destination VRF Membership

- Both ends of the tunnel must reside within the same VRF.
- The VRF associated with the **tunnel vrf** command is the same as the VRF associated with the physical interface over which the tunnel sends packets (outer IP packet routing).
- The VRF associated with the tunnel by using the **ip vrf forwarding** command is the VRF that the packets are to be forwarded in as the packets exit the tunnel (inner IP packet routing).

How to Configure GRE Tunnel IP Source and Destination VRF Membership

To configure GRE Tunnel IP Source and Destination VRF Membership on the Cisco 10000 series router, perform the following configuration tasks:

- Configuring Tunnel VRF, page 24-3
- Configuring VRF-Aware VPDN Tunnels, page 24-4

Configuring Tunnel VRF

The **tunnel vrf** command enables the Tunnel VRF feature by identifying the VRF in which the tunnel destination terminates. When configuring this feature, enter the **tunnel destination** command followed by the **tunnel vrf** command as shown in the following Summary Steps.

Use the following procedure to configure tunnel VRF on the router:

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **interface tunnel** *number*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address subnet-mask*
6. **tunnel source** (*ip-address* | *type number*)
7. **tunnel destination** *ip-address* {*hostname* | *ip-address*}
8. **tunnel vrf** *vrf-name*

For more detailed information, see the *Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership* feature module, located at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

Configuring VRF-Aware VPDN Tunnels

The **vpn** command enables the VRF-Aware VPDN Tunnels feature by associating an IP address configured in a VPDN group with a VRF. This is applied to a VPDN group as shown in the following Summary Steps.

Use the following commands to configure VRF-aware VPDN tunnels on the router:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn-group** *name*
4. **request-dialin**
5. **protocol** [**l2f** | **l2tp**| **pptp**]
6. **domain** *domain-name*
7. **exit**
8. **vpn** {**vrf** vrf-*name* | **id** *vpn-id*}
9. **source-ip** *ip-address*
10. **initiate-to ip** *ip-address* [**limit** *limit-number*] [**priority** *priority-number*]
11. **exit**

Note For Cisco IOS Release 12.2(31)SB5 and later releases, when configuring VRF-aware VPDN tunnels on the Cisco 10000 series router, different tunnels can have overlapping IP addresses across VRF instances.

For more detailed information, see the *VRF-Aware VPDN Tunnels* feature module, located at the following URL:

http://www.cisco.com/en/US/products/ps6566/products_feature_guides_list.html

Configuration Examples

This section provides the following configuration examples:

- Configuration Example for Tunnel VRF, page 24-4
- Configuration Examples for VRF-Aware VPDN Tunnels, page 24-5

Configuration Example for Tunnel VRF

The following example shows how to enable the Tunnel VRF feature by specifying the **tunnel vrf** command after the **tunnel destination** command:

```
interface Tunnel 0
    ip vrf forwarding cust 1
    ip address 10.2.0.2 255.255.255.252
    ip pim sparse-dense-mode
    tunnel source Loopback1
```

```
    tunnel destination 10.16.3.1
    tunnel vrf cust2
```

Configuration Examples for VRF-Aware VPDN Tunnels

The following example shows how to enable the VRF-Aware VPDN Tunnels feature. In the example, the **vpn** command associates the IP address 172.16.1.9 with the VRF named vrf-second, which is applied to the VPDN group named group1.

```
vpdn-group group1
    request-dialin
        protocol l2tp
!
    vpn vrf vrf-second
    source-ip 172.16.1.9
    initiate-to ip 172.16.1.1
```

The following example also enables VRF-aware VPDN tunnels and associates the VRF named vpn1 with the IP address 192.64.1.4.

```
vpdn-group Test
    accept-dialin
        protocol l2tp
        virtual-template 1
    terminate-from hostname lac
    vpn vrf vpn1
    l2tp tunnel receive-window 100
    source-ip 192.64.1.4
    initiate-to ip 192.64.1.1
```

APPENDIX A

RADIUS Attributes

This appendix lists the RADIUS attributes that the Cisco 10000 series router supports in Cisco IOS Release 12.2(4)BZ1 and later releases. The following conventions are used in the tables that follow:

- Supported and tested—The attribute has been tested and the Cisco 10000 series router supports it.
- Not Supported—The Cisco 10000 series router does not support the attribute.
- Not Applicable—The attribute does not apply to the Cisco 10000 series router.

Note For more information, see the "RADIUS Attributes" appendix in the *Cisco IOS Security Configuration Guide, Release 12.2*.

RADIUS IETF Attributes

Table A-1 RADIUS IETF Attributes

Number	IETF Attribute	Status
1	User-Name	Supported and tested
2	User-Password	Supported and tested
3	CHAP-Password	Supported and tested
4	NAS-IP Address	Supported and tested
5	NAS-Port	Supported and tested
6	Service-Type	Supported and tested
7	Framed-Protocol	Supported and tested
8	Framed-IP-Address	Supported and tested
9	Framed-IP-Netmask	Supported and tested
10	Framed-Routing	Router receives this attribute and properly handles a value of 0:None. Unclear if the system properly handles a value of 3:send and listen.
11	Filter-ID	Supported and tested
12	Framed-MTU	Supported and tested

Table A-1 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Status
13	Framed-Compression	Cisco 10000 series router ignores this attribute.
14	Login-IP-Host	Not Applicable
15	Login-Service	Not Applicable
16	Login-TCP-Port	Not Applicable
18	Reply-Message	Supported and tested
19	Callback-Number	Not Applicable
20	Callback-ID	Not Applicable
22	Framed-Route	Supported and tested
23	Framed-IPX-Network	Not Applicable
24	State	Supported but not tested
25	Class	Supported and tested
26	Vendor-Specific	Supported and tested for Cisco VSA
27	Session-Timeout	Supported and tested
28	Idle-Timeout	Supported and tested
29	Termination-Action	Typically not used in DSL environment
30	Called-Station-ID	Typically not used in DSL environment
31	Calling-Station-ID	Supported and tested
32	NAS-Identifier	Supported and tested
33	Proxy-Stat	Not Applicable
34	Login-LAT-Service	Not Applicable
35	Login-LAT-Node	Not Applicable
36	Login-LAT-Group	Not Applicable
37	Framed-AppleTalk-Link	Not Applicable
38	Framed-AppleTalk-Network	Not Applicable
39	Framed-AppleTalk-Zone	Not Applicable
40	Acct-Status-Type	Supported and tested
41	Acct-Delay-Time	Supported and tested
42	Acct-Input-Octets	Supported and tested
43	Acct-Output-Octets	Supported and tested
44	Acct-Session-Id	Supported and tested
45	Acct-Authentic	Supported and tested
46	Acct-Session-Time	Supported and tested
47	Acct-Input-Packets	Supported and tested
48	Acct-Output-Packets	Supported and tested
49	Acct-Terminate-Cause	Supported and tested

Table A-1 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Status
50	Acct-Multi-Session-Id	Multilink is not supported.
51	Acct-Link-Count	Multilink is not supported.
52	Acct-Input-Gigawords	Supported and tested
53	Acct-Output-Gigawords	Supported and tested
60	CHAP-Challenge	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
61	NAS-Port-Type	Supported and tested
62	Port-Limit	Not Applicable
63	Login-LAT-Port	Not Applicable
64	Tunnel-Type	Supported on the Cisco 10000 series router but the router only supports L2TP tunnels.
65	Tunnel-Medium-Type	Supported on the Cisco 10000 series router but IP is the only medium the router currently supports.
66	Tunnel-Client-Endpoint	Supported and tested in accounting.
67	Tunnel-Server-Endpoint	Supported and tested in accounting.
68	Acct-Tunnel-Connection	Supported and tested in Cisco IOS Release 12.2(15)BX.
69	Tunnel-Password	Supported and tested in Cisco IOS Release 12.2(15)BX.
70	ARAP-Password	Not Supported
71	ARAP-Features	Not Supported
72	ARAP-Zone-Access	Not Supported
73	ARAP-Security	Not Supported
74	ARAP-Security-Data	Not Supported
75	Password-Retry	Not Supported
76	Prompt	Typically not used in DSL environment
77	Connect-Info	Supported and tested in Cisco IOS Release 12.2(15)BX.
78	Configuration-Token	Not Supported
79	EAP-Message	Not Supported
81	Tunnel-Private-Group-ID	Not Supported
82	Tunnel-Assignment-ID	Supported and tested in Cisco IOS Release 12.2(15)BX.
83	Tunnel-Preference	Supported and tested in Cisco IOS Release 12.2(15)BX.
84	ARAP-Challenge-Response	Not Supported

Table A-1 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Status
85	Acct-Interim-Interval	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
86	Acct-Tunnel-Packets-Lost	Not Supported
87	NAS-Port-ID	Supported and tested
88	Framed-Pool	Not Supported
90	Tunnel-Client-Auth-Id	Not Supported
91	Tunnel-Server-Auth-ID	Not Supported
200	IETF-Token-Immediate	Not Applicable

Vendor-Proprietary RADIUS Attributes

Table A-2 Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	Status
17	Change-Password	Typically not used in DSL environment
21	Password-Expiration	Typically not used in DSL environment
68	Tunnel-ID	Supported and tested in accounting
108	My-Endpoint-Disc-Alias	Not Applicable
109	My-Name-Alias	Not Applicable
110	Remote-FW	Not Applicable
111	Multicast-GLeave-Delay	Not Applicable
112	CBCP-Enable	Not Applicable
113	CBCP-Mode	Not Applicable
114	CBCP-Delay	Not Applicable
115	CBCP-Trunk-Group	Not Applicable
116	Appletalk-Route	Not Applicable
117	Appletalk-Peer-Mode	Not Applicable
118	Route-Appletalk	Not Applicable
119	FCP-Parameter	Not Applicable
120	Modem-PortNo	Not Applicable
121	Modem-SlotNo	Not Applicable
122	Modem-ShelfNo	Not Applicable
123	Call-Attempt-Limit	Not Applicable
124	Call-Block-Duration	Not Applicable
125	Maximum-Call-Duration	Not Applicable
126	Router-Preference	Not Applicable

Table A-2 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Status
127	Tunneling-Protocol	Not Applicable
128	Shared-Profile-Enable	Not Applicable
129	Primary-Home-Agent	Not Applicable
130	Secondary-Home-Agent	Not Applicable
131	Dialout-Allowed	Not Applicable
133	BACP-Enable	Not Applicable
134	DHCP-Maximum-Leases	Not Applicable
135	Primary-DNS-Server	Supported and tested
136	Secondary-DNS-Server	Supported and tested
137	Client-Assign-DNS	Not Applicable
138	User-Acct-Type	Not Applicable
139	User-Acct-Host	Not Applicable
140	User-Acct-Port	Not Applicable
141	User-Acct-Key	Not Applicable
142	User-Acct-Base	Not Applicable
143	User-Acct-Time	Not Applicable
144	Assign-IP-Client	Not Applicable
145	Assign-IP-Server	Not Applicable
146	Assign-IP-Global-Pool	Not Applicable
147	DHCP-Reply	Not Applicable
148	DHCP-Pool-Number	Not Applicable
149	Expect-Callback	Not Applicable
150	Event-Type	Not Applicable
151	Session-Svr-Key	Supported and tested. Enables the router to match a user session with a client request to disconnect the session.
152	Multicast-Rate-Limit	Not Applicable
153	IF-Netmask	Not Applicable
154	Remote-Addr	Not Applicable
155	Multicast-Client	Not Applicable
156	FR-Circuit-Name	Not Applicable
157	FR-LinkUp	Not Applicable
158	FR-Nailed-Grp	Not Applicable
159	FR-Type	Not Applicable
160	FR-Link-Mgt	Not Applicable
161	FR-N391	Not Applicable
162	FR-DCE-N392	Not Applicable

Table A-2 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Status
163	FR-DTE-N392	Not Applicable
164	FR-DCE-N393	Not Applicable
165	FR-DTE-N393	Not Applicable
166	FR-T391	Not Applicable
167	FR-T392	Not Applicable
168	Bridge-Address	Not Applicable
169	TS-Idle-Limit	Not Applicable
170	TS-Idle-Mode	Not Applicable
171	DBA-Monitor	Not Applicable
172	Base-Channel-Count	Not Applicable
173	Minimum-Channels	Not Applicable
174	IPX-Route	Not Applicable
175	FT1-Caller	Not Applicable
176	Backup	Not Applicable
177	Call-Type	Not Applicable
178	Group	Not Applicable
179	FR-DLCI	Not Applicable
180	FR-Profile-Name	Not Applicable
181	Ara-PW	Not Applicable
182	IPX-Node-Addr	Not Applicable
183	Home-Agent-IP-Addr	Not Applicable
184	Home-Agent-Password	Not Applicable
185	Home-Network-Name	Not Applicable
186	Home-Agent-UDP-Port	Not Applicable
187	Multilink-ID	Multilink is not supported.
188	Num-In-Multilink	Multilink is not supported.
189	First-Dest	Not Applicable
190	Pre-Input-Octets	Not Supported
191	Pre-Output-Octets	Not Supported
192	Pre-Input-Packets	Not Supported
193	Pre-Output-Packets	Not Supported
194	Maximum-Time	Typically not used in DSL environment
195	Disconnect-Cause	Supported and tested
196	Connect-Progress	Supported and tested
197	Data-Rate	Typically not used in DSL environment
198	PreSession-Time	Typically not used in DSL environment

Table A-2 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Status
199	Token-Idle	Not Applicable
201	Require-Auth	Not Applicable
202	Number-Sessions	Not Applicable
203	Authen-Alias	Not Applicable
204	Token-Expiry	Not Applicable
205	Menu-Selector	Not Applicable
206	Menu-Item	Not Applicable
207	PW-Warntime	Not Supported
208	PW-Lifetime	Typically not used in DSL environment
209	IP-Direct	Not Applicable
210	PPP-VJ-Slot-Comp	Not Supported
211	PPP-VJ-1172	Not Supported
212	PPP-Async-Map	Not Applicable
213	Third-Prompt	Not Applicable
214	Send-Secret	Typically not used in DSL environment
215	Receive-Secret	Not Supported
216	IPX-Peer-Mode	Not Applicable
217	IP-Pool-Definition	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
218	Assign-IP-Pool	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
219	FR-Direct	Not Applicable
220	FR-Direct-Profile	Not Applicable
221	FR-Direct-DLCI	Not Applicable
222	Handle-IPX	Not Applicable
223	Netware-Timeout	Not Applicable
224	IPX-Alias	Not Applicable
225	Metric	Not Applicable
226	PRI-Number-Type	Not Applicable
227	Dial-Number	Not Applicable
228	Route-IP	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
229	Route-IPX	Not Applicable
230	Bridge	Not Applicable
231	Send-Auth	Not Applicable
232	Send-Passwd	Not Applicable
233	Link-Compression	Not Supported

Table A-2 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Status
234	Target-Util	Not Supported
235	Maximum-Channels	Not Supported
236	Inc-Channel-Count	Not Supported
237	Dec-Channel-Count	Not Supported
238	Seconds-of-History	Not Supported
239	History-Weigh-type	Not Supported
240	Add-Seconds	Not Supported
241	Remove-Seconds	Not Supported
242	Data-Filter	Supported and tested
243	Call-Filter	Not Supported
244	Idle-Limit	Not Supported
245	Preempt-Limit	Not Applicable
246	Callback	Not Applicable
247	Data-Svc	Not Applicable
248	Force-56	Not Applicable
249	Billing Number	Not Applicable
250	Call-By-Call	Not Applicable
251	Transit-Number	Not Applicable
252	Host-Info	Not Applicable
253	PPP-Address	Not Applicable
254	MPP-Idle-Percent	Not Applicable
255	Xmit-Rate	Typically not used in DSL environment.

Vendor-Specific RADIUS IETF Attributes

Table A-3 Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Status
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Not Supported
26	311	11	MSCHAP-Challenge	Not Supported
VPDN Attributes				
26	9	1	12tp-busy-disconnect	Supported in Cisco IOS but not tested on the Cisco 10000 series router.

Table A-3 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Status
26	9	1	12tp-cm-local-window-size	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-drop-out-of-order	Not Supported
26	9	1	12tp-hello-interval	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-hidden-avp	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-nosession-timeout	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-tos-reflect	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-tunnel-authen	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-tunnel-password	Supported in Cisco IOS but not tested on the Cisco 10000 series router.
26	9	1	12tp-udp-checksum	Not Supported
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Not Applicable
26	9	4	Fax-Msg-Id=	Not Applicable
26	9	5	Fax-Pages	Not Applicable
26	9	6	Fax-Coverpage-Flag	Not Applicable
26	9	7	Fax-Modem-Time	Not Applicable
26	9	8	Fax-Connect-Speed	Not Applicable
26	9	9	Fax-Recipient-Count	Not Applicable
26	9	10	Fax-Process-Abort-Flag	Not Applicable
26	9	11	Fax-Dsn-Address	Not Applicable
26	9	12	Fax-Dsn-Flag	Not Applicable
26	9	13	Fax-Mdn-Address	Not Applicable
26	9	14	Fax-Mdn-Flag	Not Applicable
26	9	15	Fax-Auth-Status	Not Applicable
26	9	16	Email-Server-Address	Not Applicable
26	9	17	Email-Server-Ack-Flag	Not Applicable

Table A-3 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Status
26	9	18	Gateway-Id	Not Applicable
26	9	19	Call-Type	Not Applicable
26	9	20	Port-Used	Not Applicable
26	9	21	Abort-Cause	Not Applicable
H323 Attributes				
26	9	23	h323-remote-address	Not Applicable
26	9	24	h323-conf-id	Not Applicable
26	9	25	h323-setup-time	Not Applicable
26	9	26	h323-call-origin	Not Applicable
26	9	27	h323-call-type	Not Applicable
26	9	28	h323-connect-time	Not Applicable
26	9	29	h323-disconnect-time	Not Applicable
26	9	30	h323-disconnect-cause	Not Applicable
26	9	31	h323-voice-quality	Not Applicable
26	9	33	h323-gw-id	Not Applicable
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Not Applicable
26	9	1	data-service	Not Applicable
26	9	1	dial-number	Not Applicable
26	9	1	force-56	Not Applicable
26	9	1	map-class	Not Applicable
26	9	1	send-auth	Not Applicable
Miscellaneous Attributes				
26	9	2	Cisco-NAS-Port	Supported and tested
26	9	1	min-links	Multilink is not supported.
26	9	1	proxyacl#<n>	Not Supported
26	9	1	spi	Not Applicable
26	9	37	Cisco-Policy-Up	Supported and tested in Cisco IOS Release 12.2(15)BZ.
26	9	38	Cisco-Policy-Down	Supported and tested in Cisco IOS Release 12.2(15)BZ.
26	9	1	atm:Peak-Cell-Rate=	Supported and tested in Cisco IOS Release 12.2(15)BX.

Table A-3 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Status
26	9	1	atm:Sustainable-Cell-Rate=	Supported and tested in Cisco IOS Release 12.2(15)BX.
26	9	1	ip:vrf-id=	Supported and tested in Cisco IOS Release 12.2(16)BX1.
26	9	1	ip:ip-unnumbered=	Supported and tested in Cisco IOS Release 12.2(16)BX1.

GLOSSARY

A

AAA authentication, authorization, and accounting (pronounced “triple a”).

AAL5 ATM adaptation layer. This layer maps higher layer user data into ATM cells, making the data suitable for transport through the ATM network.

ABR Available bit rate. QoS class defined by the ATM Forum for ATM networks. ABR is used for connections that do not require timing relationships between source and destination. ABR provides no guarantees in terms of cell loss or delay, providing only best-effort service. Traffic sources adjust their transmission rate in response to information they receive describing the status of the network and its capability to successfully deliver data.

ACL Access Control List. A list kept by routers to control access to or from the router for a number of services (for example, to prevent packets with a certain IP address from leaving a particular interface on the router).

ADSL Asymmetric digital subscriber line. A digital subscriber line (DSL) technology in which the transmission of data from server to client is much faster than the transmission from the client to the server.

ATM Asynchronous Transfer Mode. International standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

authentication A security feature that allows access to information to be granted on an individual basis.

B

bandwidth The range of frequencies a transmission line or channel can carry. The greater the bandwidth, the greater the information-carrying capacity of a channel. For a digital channel this is defined in bits. For an analog channel it is dependent on the type and method of modulation used to encode the data.

BBA Broadband Aggregation.

BGP Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined in RFC 1163.

bps Bits per second. A standard measurement of digital transmission speeds.

bridge A device that connects two or more physical networks and forwards packets between them. Bridges can usually be made to filter packets, that is, to forward only certain traffic.

broadband Characteristic of any network that multiplexes independent network carriers onto a single cable. This is usually done using frequency division multiplexing (FDM). Broadband technology allows several networks to co-exist on one single cable; traffic from one network does not interfere with traffic from another because the "conversations" happen on different frequencies in the "ether" rather like the commercial radio system.

Broadband Remote Access Server Device that terminates remote users at the corporate network or Internet users at the Internet service provider (ISP) network, that provides firewall, authentication, and routing services for remote users.

broadcast A packet delivery system where a copy of a given packet is given to all hosts attached to the network. For example: Ethernet.

C

CAR Committed access rate.

CBOS Cisco Broadband Operating System. The common operating system for DSL CPE, including the Cisco 675, Cisco 675e, Cisco 676, and Cisco 677.

CBR Constant bit rate. QoS class defined by the ATM Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery.

CBWFQ Class-based WFQ. Extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class and traffic belonging to a class is directed to the queue for that class. On the Cisco 10000 series router, the CBWFQ feature allows a VAI to inherit the service policy of the VC that the VAI uses.

CEF Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns such as the Internet, on networks characterized by intensive Web-based applications, or interactive sessions.

CE router Customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers are not aware of associated VPNs.

CHAP Challenge Handshake Authentication Protocol. Security feature supported on lines using PPP encapsulation that prevents unauthorized access. CHAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines whether that user is allowed access. Compare to PAP.

CIR Committed information rate. The reserved bandwidth for the queue. The rate at which a Frame Relay network agrees to transfer information under normal conditions, averaged over a minimum increment of time. CIR, measured in bits per second, is one of the key negotiated tariff metrics.

class-based WFQ See CBWFQ.

CoS Class of service. The three most significant bits (the User Priority bits) of the 2-byte Tag Control Information field in the IEEE 802.1p portion of a Layer 2 IEEE 802.1Q frame header. QoS uses the User Priority bits for Layer 2 CoS information. IEEE 802.1p class of service-based packet matching and marking feature enables the Cisco 10000 series router to interoperate with switches to deliver end-to-end QoS. The IEEE 802.1p standard allows QoS to classify inbound Ethernet packets based on the value in the CoS field and to explicitly set the value in the CoS field of outbound packets.

CPE Customer premises equipment. Refers to equipment located in a user's premises.

D

DBS Dynamic Bandwidth Selection. DBS dynamically changes ATM traffic shaping parameters based on a subscriber's RADIUS profile. Using this feature, wholesale service providers can sell different levels of service to retail service providers, based on the bandwidth of the ATM VC connection. The retail service provider can then offer subscribers the ability to choose services with varying levels of bandwidth allocation.

DF bit Don't Fragment indicator bit. A bit in an encapsulated header that indicates whether a router is allowed to fragment a packet.

DHCP Dynamic Host Configuration Protocol. Provides a mechanism for allocating IP addresses dynamically so that addresses can be re-used when hosts no longer need them.

Dialed Number Identification Service See DNIS.

DNIS Dialed Number Identification Service. The called party number. Typically, this is a number used by call centers or a central office where different numbers are each assigned to a specific service.

DNS Domain Name Server. The part of the distributed database system for resolving a fully qualified domain name into the four-part IP number used to route communications across the Internet.

downstream rate The line rate for return messages or data transfers from the network machine to the user's customer premises machine.

DSCP Differentiated Services Code Point

DSL Digital Subscriber Line.

DSLAM Digital Subscriber Line Access Multiplexer. Concentrates and multiplexes signals at the telephone service provider location to the broader wide area network.

Dynamic Bandwidth Selection See DBS.

E

eiBGP External and Internal Border Gateway Protocol.

encapsulation The technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer above.

Ethernet One of the most common local area network (LAN) wiring schemes, Ethernet has a transmission rate of 10, 100, or 1000 Mbps.

F

Fast switching Cisco feature whereby a route cache is used to expedite packet switching through a router.

FCC Federal Communications Commission. A U.S. government agency that regulates interstate and foreign communications. The FCC sets rates for communication services.

FTP File Transfer Protocol. The Internet protocol used to transfer files between hosts.

G

GE Gigabit Ethernet.

GRE Generic Route Encapsulation. A method of encapsulating any network protocol in another protocol.

H

high VC count Also called high VC mode. A technique used to optimize processes for session scaling.

HGW Home Gateway. Also known as L2TP Network Server (LNS) in L2TP contexts.

hop count A measure of distance between two points on the Internet. It is equivalent to the number of gateways that separate the source and destination.

HTML Hypertext Markup Language. The page-coding language for the World Wide Web.

http Hypertext Transfer Protocol. The protocol used to carry world-wide web (www) traffic between a www browser computer and the www server being accessed.

I

ICMP Internet Control Message Protocol. The protocol used to handle errors and control messages at the IP layer. ICMP is actually part of the IP protocol.

IETF Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards.

IGMP Internet Group Management Protocol. Used by IP hosts to report their multicast group memberships to an adjacent multicast router.

Internet A collection of networks interconnected by a set of routers that allow them to function as a single, large virtual network.

Internet Protocol (IP) The network layer protocol for the Internet protocol suite.

IRB Integrated routing and bridging. A protocol that allows a router to act as both bridge and router on the same interface. For broadband aggregation, we recommend using the routed bridge encapsulation (RBE) protocol. See RBE.

IP See Internet Protocol.

ISO International Standards Organization. A voluntary, non-treaty organization founded in 1946, responsible for creating international standards in many areas, including computers and communications.

ISP Internet service provider. A company that allows home and corporate users to connect to the Internet.

ITU-T International Telecommunications Union, Standardization Sector. ITU-T is the telecommunication standardization sector of ITU and is responsible for making technical recommendations about telephone and data (including fax) communications systems for service providers and suppliers.

L

L2F Layer 2 Forwarding. Protocol that supports the creation of secure virtual private dial-up networks over the Internet.

L2TP Layer 2 Tunnel Protocol. An Internet Engineering Task Force (IETF) standards track protocol defined in RFC 2661 that provides tunneling of PPP. Based upon the best features of L2F and PPTP, L2TP provides an industry-wide interoperable method of implementing VPDN.

LAC L2TP access concentrator. A node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP network server (LNS). The LAC sits between an LNS and a remote system and forwards packets to and from each. Packets sent from the LAC to the LNS requires tunneling with the L2TP protocol as defined in this document. The connection from the LAC to the remote system is either local or a PPP link.

LAN Local area network. A limited distance (typically under a few kilometers or a couple of miles) high-speed network (typically 4 to 100 Mbps) that supports many computers.

LCP Link control protocol. Protocol that establishes, configures, and tests data-link connections for use by PPP.

LNS L2TP network server. A node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP access concentrator (LAC). The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the LAC. Analogous to the Layer 2 Forwarding (L2F) home gateway (HGW).

local address pools Locally configured pools of IP addresses that the virtual home gateway (VHG) or PE router uses to assign addresses to the remote users of the PPP sessions it terminates.

M

MAC Media Access Control Layer. A sublayer of the Data Link Layer (Layer 2) of the ISO OSI Model responsible for media control.

MIB Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as SNMP or CMIP (Common Management Information Protocol). The value of a MIB object can be changed or retrieved using SNMP commands, usually through a Network Management System (NMS). MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

Modular QoS Command-line interface See MQC.

MPLS Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MPLS VPN MPLS-based virtual private network.

MQC Modular QoS Command-line interface. Also referred to as Modular CLI. A platform independent CLI for configuring QoS features on Cisco products.

MR-APS Multirouter automatic protection switching.

multicast Single packets copied by the network and sent to a specific subset of network addresses. These addresses are specified in the Destination Address Field.

multihop A term used in Cisco VPN environments. Refers to accepting a PPP session from L2TP, PPTP, or L2F and tunneling it back out using L2TP, PPTP, or L2F. See also tunnel switch.

multipoint subinterface Multipoint networks have three or more routers in the same subnet. For Dynamic Bandwidth Selection, if you put the PVC in a point-to-multipoint subinterface or in the main interface (which is multipoint by default), you need to either configure a static mapping or enable inverse Address Resolution Protocol (ARP) for dynamic mapping.

multiplexer A device that can send several signals over a single line. The signals are then separated by a similar device at the other end of the link. This can be done in a variety of ways: time division multiplexing, frequency division multiplexing, and statistical multiplexing. Multiplexers are also becoming increasingly efficient in terms of data compression, error correction, transmission speed, and multi-drop capabilities.

N

NAS Network access server. Cisco platform (or collection of platforms) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, PSTN).

NetFlow A Cisco-proprietary IP statistics collection feature that collects information on IP flows passing through a router.

NVRAM Non-Volatile Random Access Memory. The router uses this memory to store configuration information. The contents of this memory are not lost after a reboot or power cycle of the unit.

O

OAP Overlapping Address Pool. An IP address group that supports multiple IP address spaces and still allows for the verification of nonoverlapping IP address pools within a pool group.

ODAP A block of addresses managed by a central server such as a Radius server or DHCP server. Each pool is divided into subnets of various sizes. The server assigns the subnets to PE routers upon request.

on-demand address pool See ODAP.

OSI Open Systems Interconnection. An international standardization program to facilitate communications among computers from different manufacturers.

overlapping address pool See OAP.

P

PAP Password Authentication Protocol. Authentication protocol that allows PPP peers to authenticate one another. The remote router attempting to connect to the local router is required to send an authentication request. Unlike CHAP, PAP passes the password and host name or user name in the clear (unencrypted). PAP does not itself prevent unauthorized access, but merely identifies the remote end. The router or access server then determines if that user is allowed access. PAP is supported only on PPP lines. Compare with CHAP.

PCR Peak cell rate. Parameter defined by the ATM Forum for ATM traffic management.

permanent virtual circuit A fixed virtual circuit between two users. The public data network equivalent of a leased line. No call setup or clearing procedures are needed.

PE router Provider edge router. A router that is part of a service provider's network connected to a customer edge (CE) router. All VPN processing occurs in the PE router.

point-to-point subinterface With point-to-point subinterfaces, each pair of routers has its own subnet. If you put the PVC on a point-to-point subinterface, the router assumes that there is only one point-to-point PVC configured on the subinterface. Therefore, any IP packets with a destination IP address in the same subnet are forwarded on this VC. This is the simplest way to configure the mapping and is, therefore, the recommended method.

PPP Point-to-Point Protocol. The successor to SLIP, PPP provides router-to-router and host-to-network connections over both synchronous and asynchronous circuits.

PPPoA PPP over ATM. Enables a high-capacity central site router with an Asynchronous Transfer Mode (ATM) interface to terminate multiple remote PPP connections.

PPPoE PPP over Ethernet. Allows a PPP session to be initiated on a simple bridging Ethernet connected client. Refers to a signaling protocol defined within PPPoE as well as the encapsulation method. See also RFC 2516.

PPPoEoA PPP over Ethernet over ATM. Allows tunneling and termination of PPP sessions over Ethernet links and allows for Ethernet PPP connections over ATM links.

PPPoEoE PPP over Ethernet over on Ethernet. Allows tunneling and termination of PPP sessions over Ethernet links and allows for Ethernet PPP connections over Ethernet links.

PPPoEo802.1Q VLAN PPP over Ethernet over IEEE 802.1Q VLANs. Allows tunneling and termination of Ethernet PPP sessions across VLAN links. IEEE 802.1Q encapsulation is used to interconnect a VLAN-capable router with another VLAN-capable networking device. The packets on the 802.1Q link contain a standard Ethernet frame and the VLAN information associated with that frame.

PPPoX PPP over PPPoA or PPPoE or both.

PQ Priority Queuing.

PTA PPP termination aggregation. A method of aggregating IP traffic by terminating PPP sessions and aggregating the IP traffic into a single routing domain.

PTA-MD PTA Multi-Domain. A method of aggregating IP traffic by terminating PPP sessions and aggregating the IP traffic into a VPN or multiple IP routing domains.

PVC Permanent virtual circuit or connection. Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection. Compare with SVC. See also virtual circuit (VC).

PVP Permanent virtual path. Virtual path that consists of PVCs.

PXF Parallel Express Forwarding. Also referred to as *fast forwarder.* A pipelined, multiprocessor parallel packet engine, optimized for fast packet forwarding.

Q

QoS Quality of service. Cisco IOS QoS technology lets complex networks control and predictably service a variety of networked applications and traffic types.

R

RADIUS Remote Authentication Dial-In User Service (RADIUS). A client/server security protocol created by Livingston Enterprises. Security information is stored in a central location, known as the RADIUS server.

RADIUS accounting client Permits system administrators to track dial-in use.

RADIUS security client Controls access to specific services on the network.

RBE Routed bridge encapsulation. The process by which a stub-bridged segment is terminated on a point-to-point routed interface. Specifically, the router is routing on an IEEE 802.3 or Ethernet header carried over a point-to-point protocol such as PPP, RFC 1483 ATM, or RFC 1490 Frame Relay.

RD Route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 prefix.

RIP Routing Information Protocol. An IGP used to exchange routing information within an autonomous system, RIP uses hop count as a routing metric.

route The path that network traffic takes from its source to its destination. The route a datagram follows can include many gateways and many physical networks. In the Internet, each datagram is routed separately.

router A system responsible for making decisions about which of several paths network (or Internet) traffic will follow. To do this, it uses a routing protocol to gain information about the network and algorithms to choose the best route based on several criteria known as “routing metrics.”

routing table Information stored within a router that contains network path and status information. It is used to select the most appropriate route to forward information along.

S

SCR Sustainable cell rate. Parameter defined by the ATM Forum for ATM traffic management. For VBR connections, SCR determines the long-term average cell rate that can be transmitted.

SNMP Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security, typically through the use of a network management system.

SVC Switched virtual circuit. Virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic. Called a switched virtual connection in ATM terminology. Compare with PVC.

T

ToS Type of service. First defined in RFC 791.

trap Message sent by an SNMP agent to a network management station, console, or terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

tunnel A virtual pipe between the LAC and LNS that can carry multiple L2TP sessions.

tunnel switch A term used in DSL environments. Refers to a device that accepts a PPP session from L2TP, PPTP, or L2F and tunnels it again using L2TP, PPTP, or L2F. See also multihop.

turbo access control list A function of the PXF pipeline that determines whether a packet matches a list in a fixed, predictable period of time, usually regardless of the number of entries in a list. Turbo ACLs enable more expedient packet classification and access checks when the router is evaluating ACLs. The Turbo ACL feature compiles the ACLs into a set of lookup tables, while maintaining the first match requirements. Packet headers are used to access these tables in a small, fixed number of lookups, independently of the existing number of ACL entries.

U

UBR Unspecified bit rate. QoS class defined by the ATM Forum for ATM networks. UBR allows any amount of data up to a specified maximum to be sent across the network, but there are not guarantees in terms of cell loss rate and delay.

UNI signaling User Network Interface signaling for ATM communications.

upstream rate The line rate for message or data transfer from the source machine to a destination machine on the network.

V

VAI Virtual Access Interface.

VBR Variable Bit Rate. QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real time (rt) class and non-real time (nrt) class. See also VBR-nrt and VBR-rt.

VBR-nrt Variable Bit Rate-non-real time. QoS class defined by the ATM Forum for ATM networks. VBR-nrt is used for connections in which there is no fixed timing relationship between samples, but that still need a guaranteed QoS.

VBR-rt Variable Bit Rate-real time. QoS class defined by the ATM Forum for ATM networks. VBR-rt is used for connections in which there is a fixed timing relationship between samples.

VC Virtual Circuit. Also referred to as Virtual Channel. Used in ATM applications. A link that seems and behaves like a dedicated point-to-point line or a system that delivers packets in sequence, as happens on an actual point-to-point network. In reality, the data is delivered across a network via the most appropriate route. The sending and receiving devices do not have to be aware of the options and the route is chosen only when a message is sent. There is no pre-arrangement, so each virtual circuit exists only for the duration of that one transmission.

VCI Virtual channel identifier. A 16-bit field in the header of an ATM cell. The VCI, together with the VPI, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next network VCL that a cell needs to transmit on its way to its final destination. The function of the VCI is similar to that of the DLCI in Frame Relay.

VLAN Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VPDN Virtual Private Dialup Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPDNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the home gateway.

VPI Virtual path identifier. An 8-bit field in the header of an ATM cell. The VPI, together with the VCI, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next VCL that a cell needs to transmit on its way to its final destination. The function of the VPI is similar to that of the DLCI in Frame Relay.

VPN Virtual private network. A secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone. VPNs enable IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

vpnv4 Used as a keyword in commands to indicate VPN-IPv4 prefixes. These prefixes are customer VPN addresses, each of which has been made unique by the addition of an 8-byte route distinguisher.

VRF Virtual routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

VSA Vendor-Specific Attribute. An attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

W

WAN Wide area network. A data communications network that spans any distance and is usually provided by a public carrier (such as a telephone company or service provider).

WFQ Weighted Fair Queuing. A QoS congestion management function.

WRED Weighted Random Early Detection. A QoS congestion avoidance function.

X

xDSL Various types of digital subscriber lines. Examples include ADSL, HDLS, and VDSL.

INDEX

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X