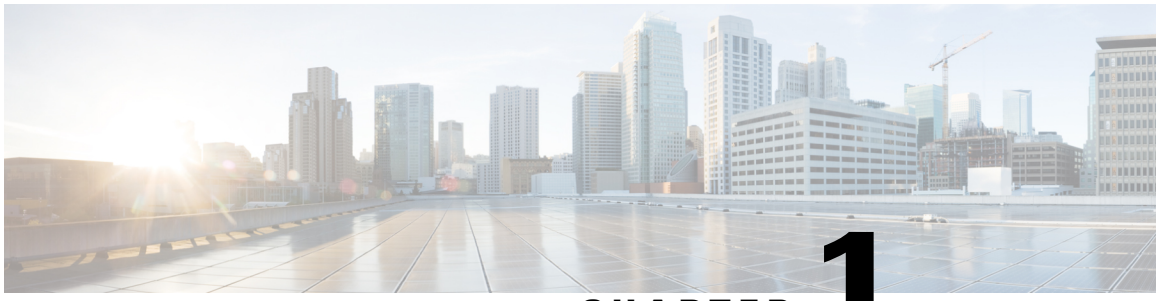
Wide-Area Networking Configuration Guide: Layer 2 Services, Cisco IOS Release 15S

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

CONTENTS



CHAPTER 1

Wide-Area Networking Overview

Cisco IOS software provides a range of wide-area networking capabilities to fit almost every network environment need. Cisco offers cell relay via the Switched Multimegabit Data Service (SMDS), circuit switching via ISDN, packet switching via Frame Relay, and the benefits of both circuit and packet switching via Asynchronous Transfer Mode (ATM). LAN emulation (LANE) provides connectivity between ATM and other LAN types. The *Cisco IOS Wide-Area Networking Configuration Guide* presents a set of general guidelines for configuring the following software components:

This module gives a high-level description of each technology. For specific configuration information, see the appropriate module.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Frame Relay

The Cisco Frame Relay implementation currently supports routing on IP, DECnet, AppleTalk, XNS, Novell IPX, CLNS, Banyan VINES, and transparent bridging.

Although Frame Relay access was originally restricted to leased lines, dialup access is now supported. For more information, for dialer profiles or for legacy dial-on-demand routing (DDR) see the see the module Dial-on-Demand Routing Configuration.

To install software on a new router or access server by downloading software from a central server over an interface that supports Frame Relay, see the module Loading and Maintaining System Images.

To configure access between Systems Network Architecture (SNA) devices over a Frame Relay network, see the module Configuring SNA Frame Relay Access Support.

The Frame Relay software provides the following capabilities:

- Support for the three generally implemented specifications of Frame Relay Local Management Interfaces (LMIs):
  - The Frame Relay Interface joint specification produced by Northern Telecom, Digital Equipment Corporation, StrataCom, and Cisco Systems
  - The ANSI-adopted Frame Relay signal specification, T1.617 Annex D
  - The ITU-T-adopted Frame Relay signal specification, Q.933 Annex A
- Conformity to ITU-T I-series (ISDN) recommendation as I122, "Framework for Additional Packet Mode Bearer Services":
  - The ANSI-adopted Frame Relay encapsulation specification, T1.618
  - The ITU-T-adopted Frame Relay encapsulation specification, Q.922 Annex A
- Conformity to Internet Engineering Task Force (IETF) encapsulation in accordance with RFC 2427, except bridging.
- Support for a keepalive mechanism, a multicast group, and a status message, as follows:
  - The keepalive mechanism provides an exchange of information between the network server and the switch to verify that data is flowing.
  - The multicast mechanism provides the network server with a local data-link connection identifier (DLCI) and a multicast DLCI. This feature is specific to our implementation of the Frame Relay joint specification.
  - The status mechanism provides an ongoing status report on the DLCIs known by the switch.
- Support for both PVCs and SVCs in the same sites and routers.

SVCs allow access through a Frame Relay network by setting up a path to the destination endpoints only when the need arises and tearing down the path when it is no longer needed.

- Support for Frame Relay Traffic Shaping beginning with Cisco IOS Release 11.2. Traffic shaping provides the following:
  - Rate enforcement for individual circuits--The peak rate for outbound traffic can be set to the committed information rate (CIR) or some other user-configurable rate.
  - Dynamic traffic throttling on a per-virtual-circuit basis--When backward explicit congestion notification (BECN) packets indicate congestion on the network, the outbound traffic rate is automatically stepped down; when congestion eases, the outbound traffic rate is stepped up again.

    - Enhanced queueing support on a per-virtual circuit basis--Custom queueing, priority queueing, and weighted fair queueing can be configured for individual virtual circuits.

- Transmission of congestion information from Frame Relay to DECnet Phase IV and CLNS. This mechanism promotes forward explicit congestion notification (FECN) bits from the Frame Relay layer to upper-layer protocols after checking for the FECN bit on the incoming DLCI. Use this Frame Relay congestion information to adjust the sending rates of end hosts. FECN-bit promotion is enabled by default on any interface using Frame Relay encapsulation. No configuration is required.
- Support for Frame Relay Inverse ARP as described in RFC 1293 for the AppleTalk, Banyan VINES, DECnet, IP, and IPX protocols, and for native hello packets for DECnet, CLNP, and Banyan VINES. It allows a router running Frame Relay to discover the protocol address of a device associated with the virtual circuit.
- Support for Frame Relay switching, whereby packets are switched based on the DLCI--a Frame Relay equivalent of a Media Access Control (MAC)-level address. Routers are configured as a hybrid DTE switch or pure Frame Relay DCE access node in the Frame Relay network.

Frame Relay switching is used when all traffic arriving on one DLCI can be sent out on another DLCI to the same next-hop address. In such cases, the Cisco IOS software need not examine the frames individually to discover the destination address, and, as a result, the processing load on the router decreases.

The Cisco implementation of Frame Relay switching provides the following functionality:

-
    - Switching over an IP tunnel
    - Switching over Network-to-Network Interfaces (NNI) to other Frame Relay switches
    - Local serial-to-serial switching
    - Switching over ISDN B channels
    - Traffic shaping on switched PVCs
    - Congestion management on switched PVCs
    - Traffic policing on User-Network Interface (UNI) DCE
    - FRF.12 fragmentation on switched PVCs

- Support for *subinterfaces* associated with a physical interface. The software groups one or more PVCs under separate subinterfaces, which in turn are located under a single physical interface. See the Configuring Frame Relay module.
- Support for fast-path transparent bridging, as described in RFC 1490, for Frame Relay encapsulated serial and High-Speed Serial Interfaces (HSSIs) on all platforms.
- Support of the Frame Relay DTE MIB specified in RFC 1315. However, the error table is not implemented. To use the Frame Relay MIB, refer to your MIB publications.
- Support for Frame Relay fragmentation. Cisco has developed the following three types of Frame Relay fragmentation:
    - End-to-End FRF.12 Fragmentation

FRF.12 fragmentation is defined by the FRF.12 Implementation Agreement. This standard was developed to allow long data frames to be fragmented into smaller pieces (fragments) and interleaved with real-time frames.

End-to-end FRF.12 fragmentation is recommended for use on PVCs that share links with other PVCs that are transporting voice and on PVCs transporting Voice over IP (VoIP).

- • Frame Relay Fragmentation Using FRF.11 Annex C

When VoFR (FRF.11) and fragmentation are both configured on a PVC, the Frame Relay fragments are sent in the FRF.11 Annex C format. This fragmentation is used when FRF.11 voice traffic is sent on the PVC, and it uses the FRF.11 Annex C format for data.

See the module Configuring Voice over Frame Relay in the *Cisco IOS Voice, Video, and Fax Configuration Guide* for configuration tasks and examples for Frame Relay fragmentation using FRF.11 Annex C.

- • Cisco Proprietary Fragmentation

Cisco proprietary fragmentation is used on data packets on a PVC that is also used for voice traffic.

See the module Configuring Voice over Frame Relay in the *Cisco IOS Voice, Video, and Fax Configuration Guide* for configuration tasks and examples for Cisco proprietary fragmentation.

Frame Relay-ATM Internetworking

Cisco IOS software supports the Frame Relay Forum implementation agreements for Frame Relay-ATM Interworking. Frame Relay-ATM Interworking enables Frame Relay and ATM networks to exchange data, despite differing network protocols. There are two types of Frame Relay-ATM Interworking.

FRF.5 Frame Relay-ATM Network Interworking

FRF.5 provides network interworking functionality that allows Frame Relay end users to communicate over an intermediate ATM network that supports FRF.5. Multiprotocol encapsulation and other higher-layer procedures are transported transparently, just as they would be over leased lines.

FRF.5 describes network interworking requirements between Frame Relay Bearer Services and Broadband ISDN (BISDN) permanent virtual circuit (PVC) services.

The FRF.5 standard is defined by the Frame Relay Forum Document Number FRF.5: *Frame Relay/ATM PVC Network Interworking Implementation Agreement.* For information about which sections of this implementation agreement are supported by Cisco IOS software, see Frame Relay-ATM Interworking Supported Standards.

FRF.8 Frame Relay-ATM Service Interworking

FRF.8 provides service interworking functionality that allows a Frame Relay end user to communicate with an ATM end user. Traffic is translated by a protocol converter that provides communication among dissimilar Frame Relay and ATM equipment.

FRF.8 describes a one-to-one mapping between a Frame Relay PVC and an ATM PVC.

The FRF.8 standard is defined by the Frame Relay Forum Document Number FRF.8: *Frame Relay/ATM PVC Network Service Interworking Implementation Agreement.* For information about which sections of this implementation agreement are supported by Cisco IOS software, see Frame Relay-ATM Interworking Supported Standards.

Switched Multimegabit Data Service

The Cisco implementation of the SMDS protocol is based on cell relay technology as defined in the Bellcore Technical advisories, which are based on the IEEE 802.6 standard. We provide an interface to an SMDS network using DS1 or DS3 high-speed transmission facilities. Connection to the network is made through a device called an SDSU--an SMDS digital service unit (DSU). The SDSU attaches to a router or access server through a serial port. On the other side, the SDSU terminates the line.

The implementation of SMDS supports the IP, DECnet, AppleTalk, XNS, Novell IPX, Banyan VINES, and OSI internetworking protocols, and transparent bridging.

The implementation of SMDS also supports SMDS encapsulation over an ATM interface. For more information and for configuration tasks, see Configuring ATM.

Routing of AppleTalk, DECnet, IP, IPX, and ISO CLNS is fully dynamic; that is, the routing tables are determined and updated dynamically. Routing of the other supported protocols requires that you establish a static routing table of SMDS neighbors in a user group. Once this table is set up, all interconnected routers and access servers provide dynamic routing.

Note When configuring IP routing over SMDS, you may need to make adjustments to accommodate split horizon effects. Refer to the Configuring EIGRP module for information about how Cisco software handles possible split horizon conflicts. By default, split horizon is *disabled* for SMDS networks.

The SMDS implementation includes multiple logical IP subnetworks support as defined by RFC 1209. This RFC describes routing IP over an SMDS cloud in which each connection is considered a host on one specific private network, and points to cases where traffic must transit from network to network.

The implementation of SMDS also provides the Data Exchange Interface (DXI) Version 3.2 with *heartbeat* . The heartbeat mechanism periodically generates a heartbeat poll frame.

When a multicast address is not available to a destination, pseudobroadcasting can be enabled to broadcast packets to those destinations using a unicast address.

Link Access Procedure - Balanced and X.25

X.25 is one of a group of specifications published by the ITU-T. These specifications are international standards that are formally called *Recommendations* . The ITU-T *Recommendation X.25* defines how connections between DTE and DCE are maintained for remote terminal access and computer communications. The X.25 specification defines protocols for two layers of the Open Systems Interconnection (OSI) reference model. The data link layer protocol defined is LAPB. The network layer is sometimes called the packet level protocol (PLP), but is commonly (although less correctly) referred to as the X.25 protocol.

The ITU-T updates its *Recommendations* periodically. The specifications dated 1980 and 1984 are the most common versions currently in use. Additionally, the International Standards Organization (ISO) has published ISO 7776:1986 as an equivalent to the LAPB standard, and ISO 8208:1989 as an equivalent to the ITU-T 1984 *Recommendation X.25* packet layer. The Cisco X.25 software follows the ITU-T 1984 *Recommendation X.25* , except for its Defense Data Network (DDN) and Blacker Front End (BFE) operation, which follow the ITU-T 1980 *Recommendation X.25* .

Note The ITU-T carries out the functions of the former CCITT. The 1988 X.25 standard was the last published as a CCITT *Recommendation* . The first ITU-T *Recommendation* is the 1993 revision.

In addition to providing remote terminal access, The Cisco X.25 software provides transport for LAN protocols--IP, DECnet, XNS, ISO CLNS, AppleTalk, Novell IPX, Banyan VINES, and Apollo Domain--and bridging.

Cisco IOS X.25 software provides the following capabilities:

- LAPB datagram transport--LAPB is a protocol that operates at Level 2 (the data link layer) of the OSI reference model. It offers a reliable connection service for exchanging data (in units called *frames*) with one other host. The LAPB connection is configured to carry a single protocol or multiple protocols. Protocol datagrams (IP, DECnet, AppleTalk, and so forth) are carried over a reliable LAPB connection, or datagrams of several of these protocols are encapsulated in a proprietary protocol and carried over a LAPB connection. Cisco also implements transparent bridging over multiprotocol LAPB encapsulations on serial interfaces.
- X.25 datagram transport-- X.25 can establish connections with multiple hosts; these connections are called virtual circuits. Protocol datagrams (IP, DECnet, AppleTalk, and so forth) are encapsulated inside packets on an X.25 virtual circuit. Mappings between the X.25 address of a host and its datagram protocol addresses enable these datagrams to be routed through an X.25 network, thereby permitting an X.25 PDN to transport LAN protocols.
- X.25 switch--X.25 calls can be routed based on their X.25 addresses either between serial interfaces on the same router (local switching) or across an IP network to another route r, using X.25 over TCP (XOT). XOT encapsulates the X.25 packet level inside a TCP connection, allowing X.25 equipment to be connected via a TCP/IP-based network. The Cisco X.25 switching features provide a convenient way to connect X.25 equipment, but do not provide the specialized features and capabilities of an X.25 PDN.
- ISDN D channel--X.25 traffic over the D channel, using up to 9.6 kbps bandwidth, can be used to support many applications. For example, it may be required as a primary interface where low volume sporadic interactive traffic is the normal mode of operation. For information on how to configure X.25 on ISDN, refer to the modules Configuring X.25 on ISDN and Configuring X.25 on ISDN Using AO/DI.
- PAD--User sessions can be carried across an X.25 network using the packet assembler/disassembler (PAD) protocols defined by the ITU-T Recommendations X.3 and X.29.
- QLLC--The Cisco IOS software can use the Qualified Logical Link Control (QLLC) protocol to carry SNA traffic through an X.25 network.
- Connection-Mode Network Service (CMNS)--CMNS is a mechanism that uses OSI-based network service access point (NSAP) addresses to extend local X.25 switching to nonserial media (for example, Ethernet, FDDI, and Token Ring). This implementation provides the X.25 PLP over Logical Link Control, type 2 (LLC2) to allow connections over nonserial interfaces. The Cisco CMNS implementation supports services defined in ISO Standards 8208 (packet level) and 8802-2 (frame level).
- DDN and BFE X.25--The DDN-specified Standard Service is supported. The DDN X.25 Standard Service is the required protocol for use with DDN Packet-Switched Nodes (PSNs). The Defense Communications Agency (DCA) has certified the Cisco DDN X.25 Standard Service implementation for attachment to the DDN. The Cisco DDN implementation also includes Blacker Front End operation.
- X.25 MIB--Subsets of the specifications in *SNMP MIB Extension for X.25 LAPB* (RFC 1381) and *SNMP MIB Extension for the X.25 Packet Layer* (RFC 1382) are supported. The LAPB XID Table, X.25 Cleared

Circuit Table, and X.25 Call Parameter Table are not implemented. All values are read-only. To use the X.25 MIB, refer to the RFCs.

- Closed User Groups (CUGs)--A CUG is a collection of DTE devices for which the network controls access between two members and between a member and a nonmember. An X.25 network can support up to 10,000 CUGs. CUGs allow various network subscribers (DTE devices) to be segregated into private subnetworks that have limited incoming or outgoing access.

The Cisco X.25 implementation does not support fast switching.

Layer 2 Virtual Private Network

L2VPN services are point-to-point. They provide Layer 2 point-to-point connectivity over either an MPLS or a pure IP (L2TPv3) core.

Layer 2 Tunneling Protocol Version 3

The Layer 2 Tunneling Protocol Version 3 feature expands Cisco's support of Layer 2 VPNs. Layer 2 Tunneling Protocol Version 3 (L2TPv3) is an IETF l2tpext working group draft that provides several enhancements to L2TP to tunnel any Layer 2 payload over L2TP. Specifically, L2TPv3 defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network by using Layer 2 VPNs.

L2VPN Pseudowire Redundancy

L2VPNs can provide pseudowire resiliency through their routing protocols. When connectivity between end-to-end PE routers fails, an alternative path to the directed LDP session and the user data can take over. However, there are some parts of the network where this rerouting mechanism does not protect against interruptions in service. The L2VPN Pseudowire Redundancy feature provides the ability to ensure that the CE2 router in can always maintain network connectivity, even if one or all the failures in the figure occur. The L2VPN Pseudowire Redundancy feature enables you to set up backup pseudowires. You can configure the network with redundant pseudowires (PWs) and redundant network elements.

Layer 2 Virtual Private Network 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 the different Layer 2 encapsulations. The L2VPN Interworking feature supports Ethernet, 802.1Q (VLAN), Frame Relay, ATM AAL5, and PPP attachment circuits over MPLS and L2TPv3.

Layer 2 Local Switching

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) or between interfaces of different types (for example, Frame Relay to ATM) on the same router. The interfaces can be on the same line card or on two different cards.

During these kinds of switching, the Layer 2 address is used, not any Layer 3 address. Same-port local switching allows you to switch Layer 2 data between two circuits on the same interface.

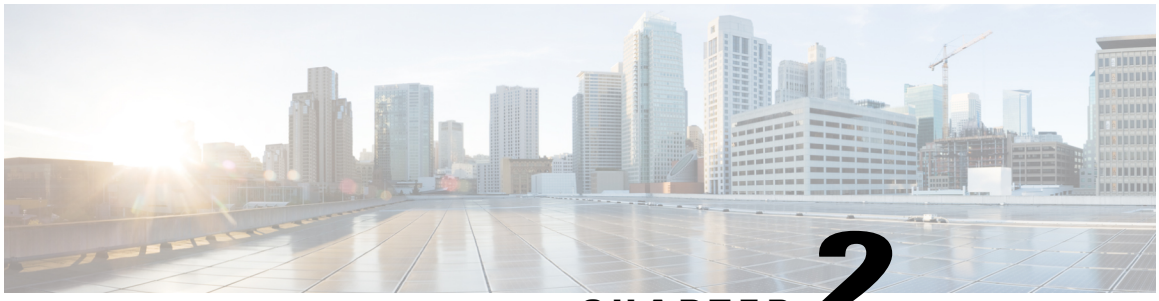
Wide Area Application Services

Cisco's WAAS Express software interoperates with WAN optimization headend applications from Cisco and improves WAN access and use by optimizing applications that require high bandwidth or are bound to a LAN, such as backup.

WAAS Express helps enterprises meet the following objectives:

- Complements the Cisco WAN optimization system by adding the capability to the branch routers.
- Provide branch office employees with LAN-like access to information and applications across a geographically distributed network.
- Minimize unnecessary WAN bandwidth consumption through the use of advanced compression algorithms.
- Virtualize print and other local services to branch office users.
- Improve application performance over the WAN by addressing the following common issues:
    - Low data rates (constrained bandwidth)
    - Slow delivery of frames (high network latency)
    - Higher rates of packet loss (low reliability)

The Network Analysis Module (NAM) Performance Agent (PA) for WAAS Express analyzes and measures network traffic. The PA enables baselining, monitoring, and troubleshooting of application performance. The analysis and measurement of network traffic is done by the Measurement, Aggregation, and Correlation Engine (MACE). MACE performs the required measurements on a subset of traffic and exports the necessary metrics to a target.



CHAPTER 2

Layer 2 Tunneling Protocol Version 3

The Layer 2 Tunneling Protocol Version 3 feature expands Cisco's support of Layer 2 VPNs. Layer 2 Tunneling Protocol Version 3 (L2TPv3) is an IETF l2tpext working group draft that provides several enhancements to L2TP to tunnel any Layer 2 payload over L2TP. Specifically, L2TPv3 defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network by using Layer 2 VPNs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Layer 2 Tunneling Protocol Version 3

- Before you configure an xconnect attachment circuit for a provider edge (PE) device (see the Configuring the Xconnect Attachment Circuit task), the Cisco Express Forwarding (formerly known as CEF) feature

must be enabled. To enable Cisco Express Forwarding on an interface, use the **ip cef** or **ip cef distributed** command.

- You must configure a loopback interface on the router for originating and terminating the L2TPv3 traffic. The loopback interface must have an IP address that is reachable from the remote PE device at the other end of an L2TPv3 control channel.

Restrictions for Layer 2 Tunneling Protocol Version 3

General L2TPv3 Restrictions

- Cisco Express Forwarding must be enabled for the L2TPv3 feature to function. The xconnect configuration mode is blocked until Cisco Express Forwarding is enabled. On distributed platforms, such as the Cisco 7500 series, if Cisco Express Forwarding is disabled while a session is established, the session is torn down. The session remains down until Cisco Express Forwarding is reenabled. To enable Cisco Express Forwarding, use the **ip cef** or **ip cef distributed** command.
- The number of sessions on PPP, High-Level Data Link Control (HDLC), Ethernet, or 802.1q VLAN ports is limited by the number of interface descriptor blocks (IDBs) that the router can support. For PPP, HDLC, Ethernet, and 802.1q VLAN circuit types, an IDB is required for each circuit.
- When L2TPv3 is used to tunnel Frame Relay D channel data-link connection identifiers (DLCIs), an IDB is not required for each circuit. As a result, the memory requirements are much lower. The scalability targets for the Engineering Field Test (EFT) program are 4000 L2TP session.
- To convert an interface with Any Transport over MPLS (AToM) xconnect to L2TPv3 xconnect, remove the AToM configuration from the interface and then configure L2TPv3. Some features may not work if L2TPv3 is configured before removing the AToM configuration.
- Frame Relay support includes only 10-bit DLCI addressing. The L2TPv3 feature does not support Frame Relay extended addressing.
- The interface keepalive feature is automatically disabled on the interface to which xconnect is applied, except for Frame Relay encapsulation, which is required for Local Management Interface (LMI).
- Static L2TPv3 sessions do not support Frame Relay LMI interworking.
- Static L2TPv3 sessions do not interoperate with Universal Tunnel Interface (UTI) using keepalives.
- Layer 2 fragmentation of IP packets and Intermediate System-to-Intermediate System (IS-IS) fragmentation through a static L2TPv3 session are not supported.
- Layer 3 fragmentation is not recommended because of performance degradation.
- The L2TPv3 Layer 2 (IP packet) fragmentation feature (see the Configuring the L2TPv3 Pseudowire task) is not supported when the customer edge (CE) router is running special Layer 2 options such as Layer 2 sequencing, compression, or encryption. Examples of these options are Frame Relay compression and fragmentation or PPP compression. In these scenarios, the IP payload is not in a format that is compatible with IP fragmentation.
- The Stateful Switchover (SSO), Route Processor Redundancy (RPR) and RPR+ components of the HA functions are supported only at the coexistence level. If you attempt a switchover using SSO, RPR, or

RPR+, the tunnels will fail and then eventually recover after an undetermined time duration. This includes both IPv4 and IPv6 traffic.

- Interworking is not allowed when sequencing is enabled.
- Untagged packets (native VLAN) forwarding for xconnect that is configured on the dot1q subinterface is not supported.
- L2TPv3 xconnect is not supported on an EtherSwitch module. This limitation is also applicable to switch virtual interfaces (SVI) that are physically terminated on an EtherSwitch module interface.

Cisco 7200 Series and Cisco 7301 Specific Restrictions

- ATM port mode cell relay is only supported on the PA-A3-T3, PA-A3-E3, and PA-A3-OC-3 ATM port adapters.
- The features ATM Single Cell Relay VC Mode over L2TPv3 and ATM VP Mode Single Cell Relay over L2TPv3 are only supported on the PA-A3-T3, PA-A3-E3, and PA-A3-OC-3 ATM port adapters.
- VPI or VPI/VCI rewrite is not supported for any ATM transport mode. Both pairs of PE to CE peer routers must be configured with matching VPI and VCI values except in OAM local emulation mode. For example, if PE1 and CE1 are connected by PVC 10/100, PE2 and CE2 should also be connected by PVC 10/100.
- In OAM local emulation mode only, the VPI/VCI values used for each pair of PE to CE routers need not match. PE1 and CE1 may be configured with one VPI/VCI value, and PE2 and CE2 may be configured with a different VPI/VCI value. For example, if PE1 and CE1 are connected by PVC 10/100, PE2 and CE2 may be connected by PVC 20/200.

Cisco 7304 Specific Restrictions

- The L2TPv3 Distributed Sequencing feature in Cisco IOS Release 12.2(27)SBC is supported only on the Cisco 7304 NPE-G100.
- The Protocol Demultiplexing feature in Cisco IOS Release 12.2(27)SBC is supported only on the Cisco 7304 NPE-G100.
- On the Cisco 7304 platforms, ATM cell relay is supported only on the PA-A3-T3, PA-A3-E3, and PA-A3-OC-3 ATM port adapters. ATM cell relay is not supported on the native line cards 7300-1OC-12ATM and 7300-2OC-3ATM.

Cisco 7500 Series-Specific Restrictions

- Distributed sequencing is supported on Cisco 7500 series routers only. The **ip cef distributed**command must be configured.
- ATM port mode cell relay is supported only on the PA-A3-T3, PA-A3-E3, and PA-A3-OC-3 ATM port adapters.
- VPI or VPI/VCI rewrite is not supported for any ATM transport mode. The peer routers must be configured with matching VPI or VCI values.

Supported Shared Port Adapters for the Cisco 7600 Series Router

The following shared port adapters (SPAs) support L2TPv3 on the Cisco 7600 series routers.

Ethernet

- SPA_TYPE_ETHER_2xGE (2-port Gigabit Ethernet)
- SPA_TYPE_ETHER_2xGE_V2 (2-port Gigabit Ethernet)
- SPA_TYPE_ETHER_5xGE_V2 (5-port Gigabit Ethernet)
- SPA_TYPE_ETHER_1x10GE_V2 (single-port 10-Gigabit Ethernet)

ATM

- SPA_TYPE_KATM_2xOC3 (ATM, 2-port OC3)
- SPA_TYPE_KATM_4xOC3 (ATM, 4-port OC3)
- SPA_TYPE_KATM_1xOC12 (ATM, 1-port OC12)
- SPA_TYPE_KATM_1xOC48 (ATM, 1-port OC48)
- SPA_TYPE_CEOP_24xT1E1(CEoP 24-port T1/E1)
- SPA_TYPE_CEOP_1xOC3 (CEoP 1-port OC3)
- SPA_TYPE_CEOP_2xT3E3 (CEoP 2-port T3/E3)

Cisco 7600 Series-Specific Restrictions

On the Cisco 7600 series routers, L2TPv3 is a line card feature that was traditionally implemented only on the 7600-SIP-400 line card. In Cisco IOS Release 12.2(33)SRD, L2TPv3 is supported on the 7600-ES+20/40 line cards in the hardware, with the same capabilities (excluding the non-Ethernet interface support) and restrictions as in the 7600-SIP-400 line card. The minimum hardware requirement for enabling the L2TPv3 service on a Cisco 7600 router are an L2TPv3-aware line card (such as the 7600-SIP-400/ES+) at the Layer 2 CE-facing side and an IP interface on any line card at the IP core-facing side. A service card is not required for L2TPv3.

General Restrictions

L2TPv3 imposes the following general restrictions:

- The layer 2-facing line card must be an L2TPv3-supporting line card.
- There must be at least one distinct L2TPv3 tunnel per Layer 2-facing line card.
- Only IPv4 tunneling is supported for Layer 2 frames (configurations such as EoL2TPv3oMPLS (on the encapsulating provider edge (PE) device are not supported).

EVC/EFP Restrictions

L2TPv3 is not supported in conjunction with EVC features. L2TPv3 can coexist with EVC on the same port, meaning that while one subinterface is used to tunnel dot1q-tagged traffic over L2TP, another subinterface can be used to perform EVC features.

SVI VLAN Interfaces Restrictions

L2TPv3 is not supported on SVI VLAN interfaces.

MIB Support Restrictions

There is no L2TPv3-specific MIB support.

Layer Frame Fragmentation Restrictions

Layer 2 frame fragmentation is not supported. Even if the Layer 2 frame recovered after the L2TPv3 decapsulation exceeds the Layer 2 MTU on the CE-facing interface, the SIP-400 line card still sends the entire Layer 2 frame to the CE device. The Layer 2 frame may be dropped on the CE device because of MRU violations.

Layer 2 Virtual Private Network Interworking Restrictions

The SIP-400 line card does not support Layer 2 VPN interworking ("like to like" is the only mode supported for L2TPv3 tunneling).

Packet Sequencing Restrictions

The initial release of L2TPv3 focuses on tunneling Ethernet and ATM traffic over L2TPv3. Because of performance issues, the SIP-400 line card does not support L2TPv3 packet sequencing for Ethernet and ATM traffic. As a result, the 4-byte Layer 2-specific sublayer control word is not supported for Ethernet pseudowires. Configuring sequencing on a pseudowire will cause L2VPN traffic corruption.

By default, sequencing is disabled. However, you can configure sequencing in the pseudowire class, because the pseudowire class may be applied to pseudowires on other 7600 line cards that support sequencing. You must keep sequencing disabled when the pseudowire is handled on the SIP-400 line card.

Counters Restrictions

Per-session counters are provided by the line card. Per-tunnel counters are not provided.

Security and QoS ACLs Restrictions

The security QoS ACLs are not supported on the Layer 2 interfaces facing customer device, which means that you cannot apply ACLs to Layer 2 VPN traffic. (The Security ACL and the QoS ACL can still be applied to the IP interfaces at the core-facing side.)

DF Bit Reflection from Inner IP to Outer IP Restrictions

Traffic on ATM interfaces may have a deep stack of Layer 2 encapsulations. For example, the IP packet may be embedded first in Ethernet, then in Subnetwork Access Protocol (SNAP) and ATM Adaptation Layer 5 (AAL5). There is no guarantee that the SIP-400 line card will find the IP packet inside the AAL5 envelope. Therefore, Don't Fragment (DF) bit reflection from inner IP to outer IP is not performed for traffic on ATM interfaces.

Session Cookie

A cookie check is supported for data packets. Cookies (remote and local) can be part of the decapsulation table indexed by *session-id*.

Scalability

Up to 8000 pseudowires and 512 tunnels are supported.

Set DF Bit in Outer IP

When the **ip dfbit set** command is configured for the pseudowire, the SIP-400 line card sets the DF bit in the outer IP header during L2TPv3 encapsulation. This DF bit handling is subject to IS-IS packet fragmentation.

Set TTL in Outer IP

When the **ip ttl value** command is configured for the pseudowire, the SIP-400 line card sets the TTL value in the outer IP header during L2TPv3 encapsulation. When the TTL value is not set, the TTL value in the outer IP header is set to 254.

Layer 2-Specific Sublayer Control Word

The Layer 2-specific sublayer control word is defined in L2TPv3 RFCs solely for the purpose of packet sequencing (with the exception of AAL5 payload). On Cisco 7200 series, Cisco 7500 series, and Cisco 12000 series routers, the control word is omitted when sequencing is disabled on non-ATM AAL5 pseudowires. To interoperate with Cisco 7200 series, Cisco 7500 series, and Cisco 12000 series routers, the SIP-400 line card does not support control words on all non-AAL5 pseudowire types in the initial release.

Table 1: Layer 2 VPN over L2TPv3 Protocol Stack (without Sequencing)

L2TPv3 Packet Stack for AAL5 Payload	L2TPv3 Packet Stack for Non-AAL5 Payload
20 bytes IP header Protocol ID = 115	20 bytes IP header Protocol ID = 115
4 bytes session ID	4 bytes session ID
0, 4 or 8 bytes cookie	0, 4 or 8 bytes cookie
4 bytes control word	Layer 2 frame (non-AAL5
AAL5 frame	

MTU Support

MTU processing is done on the ingress path on the SIP-400 line card. The SIP-400 line card enforces Layer 2 MRU checking for every Layer 2 frame received from the CE device. All frames that fail MRU checking are dropped, and the accepted frames are entered into the L2TPv3 encapsulation process. During the process, the whole L2TPV3 packets (including outer IP) are checked again using IP MTU. The packets that pass IP MTU checking are sent to Enhanced Address Recognition Logic (EARL) for IP routing. The failed packets are sent to RP for IP fragmentation or for drop accounting and notifying.

Path MTU discovery is enabled when the **ip pmtu** command is configured for the pseudowire. This feature requires an ingress Layer 2 frame to be dropped if, after L2TPv3 encapsulation, the total packet length exceeds L2TP tunnel path MTU, and the DF bit of the IP header inside the Layer 2 frame is 1. To support this feature, the SIP-400 line card performs tunnel path MTU checking on each ingress Layer 2 frame during L2TPv3 encapsulation phase. If the total packet length after encapsulation exceeds path MTU, the SIP-400 line card forwards the original Layer 2 frame to the route processor. On receiving the Layer 2 frame, the route processor may send an Internet Control Message Protocol (ICMP) unreachable message to the source of the IP packet, depending on how deep the IP packet is embedded in the Layer 2 frame.

L2TPv3 IP packet fragmentation and reassembly is done by software on the route processor. The SIP-400 line card performs core-facing interface IP MTU checking on all packets encapsulated in L2TPv3. If the MTU checking fails, the original Layer 2 frames are sent to the route processor for IP fragmentation. Fragmented L2TPV3 IP packets received from the IP core are received by the route processor from the core facing interface by EARL. The route processor handles L2TPv3 packet reassembly and recovers the inner Layer 2 frame. The route processor also sends the Layer 2 frame to the CE-facing interface by using index-directed WAN dbus frames.

With IS-IS packet fragmentation, IS-IS packets are often padded to the maximum MTU size. L2TPv3 encapsulation increases the packet size by 28 to 36 bytes. A Layer 2 frame with an IS-IS packet embedded

may exceed the tunnel path MTU after L2TPV3 encapsulation. Therefore, Layer 3 fragmentation is often needed. To support fragmentation, the SIP-400 line card searches for IS-IS packets in a Layer 2 Frame. If an IS-IS packet is found during L2TPv3 encapsulation, the SIP-400 line card clears the DF bit in the outer IP and sets IP precedence to 6. This allows the IP packet to be fragmented when traveling through the IP core.

Ethernet Attachment Circuits

The SIP-400 line card supports Ethernet over L2TPv3 in compliance with RFC4719. Two types of pseudowire are supported: Ethernet VLAN pseudowire type (0x0004) and Ethernet pseudowire type (0x0005). When xconnect is configured on an Ethernet main interface, Ethernet frames are tunneled over L2TPv3 using Ethernet port pseudowires (type 0x0005). In this mode, Ethernet frames received on the port (tagged or untagged) are delivered to the remote CE device unaltered.

When xconnect is configured on a dot1q subinterface, the tagged Ethernet frames are tunneled using an Ethernet VLAN pseudowire (type 0x0004). In this case, the pseudowire connects one Ethernet VLAN to another Ethernet VLAN. Received Ethernet VLAN frames from the CE device are tunneled over L2TPv3 unchanged. When arriving on the destination PE device, the original VLAN tag is written to use the destination VLAN ID. While doing so, the priority field in the VLAN tag is preserved.

Ethernet OAM Support

The SIP-400 line card supports service-level OAM and link-level OAM features on Ethernet interfaces.

Service-level OAM packets, also known as Connectivity Fault Management (CFM) packets, are sent using SNAP header with type 0x0126. Link-level OAM packets, also known as Link Monitoring (LM) packets are sent on Ether-Type 0x8809.

The SIP-400 line card monitors the above two types of ingress OAM frames from the CE device. When the OAM frames are found and OAM features are configured on the Ethernet interface, the OAM frames are intercepted and forwarded to the route processor. If there is no Ethernet OAM configuration, all OAM frames are tunneled in L2TPv3 as normal data frames.

ATM Attachment Circuits

The SIP-400 line card supports ATM over L2TPv3 in compliance with RFC 4454 with minor deviation. RFC 4454 defines four types of ATM pseudowire:

- ATM AAL5 SDU VCC transport (0x0002)
- ATM cell transport port mode (0x0003)
- ATM cell transport VCC mode (0x0009)
- ATM cell transport VPC mode (0x000A)

ATM cell transport port mode is not supported.

When xconnect is configured on a PVC with encapsulation AAL5, ATM AAL5 pseudowire (0x0002) is used to tunnel AAL5 frames between PE devices. The SIP-400 line card supports Layer 2 sublayer-specific control words for AAL5 pseudowire. This is the only type of pseudowire allowed to carry control words.

When xconnect is configured on PVC in AAL0 mode, an ATM cell transport VCC pseudowire (type 0x0009) is used. When xconnect is configured on PVP in AAL0 mode, an ATM cell transport VPC pseudowire (type 0x000A) is used. In both types of pseudowire, each L2TPv3 packet carries one ATM cell. Cell packing is not supported.

ATM OAM Cells

The SIP-400 line card supports ATM OAM cells operating at VP and VC levels. F4 cells operate at the VP level. They use the same VPI as the user data cells. However, they use two different reserved VCIs, as follows:

- VCI = 3 Segment OAM F4 cells
- VCI = 4 End-to-end OAM F4 cells

OAM F5 cells operate at the VC level. They use the same VPI and VCI as the user cells. To distinguish between data and OAM cells, the PTI field is used as follows:

- PTI = 100 (4) Segment OAM F5 cells processed by the next segment
- PTI =101 (5) End-to-end OAM F5 cells which are only processed by end stations terminating an ATM link

In the ingress direction (CE to PE), because of OAM emulation not supported in the 12.2(33)SRC release, all OAM cells are handled the same as data cells on the SIP-400 line card. Both segment and end-to-end OAM F4/F5 cells are tunneled over L2TPv3 to the remote PE device. They are sent transparently across the IP core in L2TPv3 tunnels.

In the egress direction (PE to CE), the SIP-400 line card sends all OAM cells to the CE device similar to sending ATM data cells.

Loopback Interface Reservation

You must reserve a loopback interface used as a source of the L2TPv3 tunnel for a particular line card to prevent it from being used across multiple line cards. These loopback interfaces host the local IP addresses used by the L2TP tunnels. A minimum of one such IP address is needed for every CE-facing line card. In most cases, you must create multiple loopback interfaces to accommodate routing protocol configuration and L2TPv3 configuration. Also, you must explicitly use the **mpls ldp router-id** command to avoid LDP router ID changes after system reload.

To reserve a loopback interface, use the **mls reserve l2tpv3 slot** *slot-number* [**processor** *processor-number*] command on the route processor in interface configuration mode.

This command binds the loopback interface to the specified slot/NP. Once configured, the loopback cannot be used to configure L2TPv3 tunnels on other LC/NPs. You must create another loopback interface in order to configure an L2TPv3 pseudowire on an interface that resides on another LC/NP.

QoS

QoS is handled on the line card. EARL does not perform QoS operations on L2TPv3 packets.

QoS at L2TPv3 Tunnel Ingress

The SIP-400 line card applies QoS to ingress traffic before doing L2TPv3 encapsulation. Given the order of traffic processing, the SIP-400 line card can support full-fledged interface/PVC level MQC on Layer 2 traffic. QoS on IP tunnel traffic is limited to ToS marking only.

The supported QoS-on-ingress Layer 2 frames are as follows.

- Classification. Ethernet interfaces: match on vlan, cos, ip dscp, ip precedence. ATM interfaces: match on atm clp
- Marking:
  - Ethernet interfaces: set cos
  - ATM interfaces: none
- Policing on both Ethernet and ATM interfaces
- Queuing on Ethernet interfaces

QoS at L2TPv3 Tunnel Egress

With egress traffic flow on the SIP-400 line card, QoS is again applied to Layer 2 traffic after L2TPv3 de-encapsulation. While the SIP-400 line card can support full-fledged Layer 2 MQC at the interface/PVC level, no QoS can be done on the IP tunnel traffic.

The supported QoS-on-egress Layer 2 frames are as follows.

- Classification:
  - Ethernet interfaces: match on vlan, cos, ip dscp, ip precedence
  - ATM interfaces: none
- Marking:
  - Ethernet interfaces: set cos, ip dscp, ip precedence
  - ATM interfaces: set atm clp
- Policing on both Ethernet and ATM interfaces
- Queuing on both Ethernet and ATM interfaces

L2TPv3 Packet ToS Marking

L2TPv3 packet ToS marking is done on the SIP-400 ingress path. There are three ways of marking the ToS field:

- Configure the **ip tos value** *value* command on each pseudowire to set the ToS field
- Configure the **ip tos reflect** command on each pseudowire to allow the inner IP ToS copied to the outer IP ToS
- By default, Layer 2 QoS is automatically reflected to outer IP ToS. For example, if the Layer 2 frame is an 802.Q frame, the 3-bit priority field in the VLAN tag is copied to the precedence bits in the outer IP ToS field

When the **ip tos reflect** command is configured, the SIP-400 line card searches for an IP header inside each received Layer 2 frame. If an IP packet is found, its ToS is copied to the outer ToS. Otherwise, the ToS value in the L2TPv3 IP header is set 0.

When neither the **ip tos value** command nor the **ip tos reflect** command is configured, the SIP-400 line card searches for a VLAN tag in each Ethernet frame. If a tag is found, the inner Layer 2 QoS is reflected to the outer IP ToS. Otherwise, the L2TPv3 IP ToS field is set 0.

Cisco 10720-Specific Restrictions

- Variable cookie size and L2TPv3 sequencing are not supported.
- Starting in Cisco IOS Release 12.0(32)SY, the L2TPv3 Layer 2 Fragmentation feature is supported on the Cisco 10720 Internet router to enable the fragmentation of IP packets to occur before data enters the pseudowire. When you enable this feature in an L2TPv3 pseudowire configuration using the **ip pmtu** command, the Don't Fragment (DF) bit in the outer Layer 2 packet header is automatically set on so that (for performance reasons) tunneled packets are not reassembled on the decapsulation router.

- The Cisco 10720 Internet router supports the reassembly only of fragmented IS-IS packets in an L2TPv3 pseudowire. IS-IS packet reassembly is performed by the Route Processor (RP) at the process level, not in the Parallel eXpress Forwarding (PXF) forwarding path.
- On the Cisco 10720 Internet router, the **uti translation** command is not migrated for xconnect service and is not supported. Although the **uti** command is supported in L2TPv3 releases, the **translation** option is lost in the migration.
- On the Cisco 10720 Internet router, although it is not required, we highly recommend that you configure a loopback interface as the IP local interface.

You can also configure a LAN interface as the IP local interface so that the tunnel control session is tied to an operational LAN (Gigabit Ethernet or Fast Ethernet) interface or subinterface. However, in this case, the tunnel control plane is used only as long as the Gigabit Ethernet or Fast Ethernet interface is operational.

Cisco 12000 Series-Specific Restrictions

Tunnel Server Card Versus Native L2TPv3 Implementation

On the Cisco 12000 series Internet router, L2TPv3 is implemented in two different ways:

- The 1-port OC-48c/STM-16c POS/SDH line card is required as the dedicated tunnel server card (TSC) to accelerate the encapsulation and decapsulation of Layer 2 data on engine 2 (and earlier engine types) line cards in an L2TPv3 tunnel session.
- The enhanced edge capabilities of IP services engine (ISE) and engine 5 line cards do not require a tunnel server card for Layer 2 data encapsulation and decapsulation in an L2TPv3 tunnel. This is called a *native L2TPv3* session.

Note Native L2TPv3 tunnel sessions on customer-facing ISE and Engine 5 line cards can coexist with tunnel sessions that use a tunnel server card.

Different combinations of engine types are supported as customer-facing and backbone-facing line cards for encapsulation and decapsulation in L2TPv3 tunneling.

Note If you have native cards (engine 3 and engine 5) in the PE routers and the Tunnel Server Card is configured to support the non-native cards, then you must remove the TSC configuration by using the **no hw-module slot** *number* **mode server** command. If the TSC configuration exists in the PE router and the TSC card is removed, all the tunnels will fail.

L2TPv3 Encapsulation

When a Layer 2 packet arrives on a customer-facing interface, if the interface is bound to an L2TPv3 tunnel, L2TPv3 encapsulation is supported as follows:

- If the customer-facing line card is engine 2 or an earlier engine type, the line card forwards the packet to the tunnel server card, which performs L2TPv3 encapsulation.
- If the customer-facing line card is ISE or engine 5, the line card performs L2TPv3 encapsulation.

A backbone-facing line card of any engine type sends the packet across the service provider backbone network.

L2TPv3 Decapsulation

When an L2TPv3 packet arrives on a backbone-facing interface, L2TPv3 decapsulation is supported as follows:

- If the backbone-facing line card is non-ISE/E5 (any engine type besides ISE and Engine 5), the line card forwards the packet to the tunnel server card. The tunnel server card determines if the packet is bound to an Engine 2 (or earlier engine) or an ISE/E5 customer-facing line card.
  - If the packet is bound to an Engine 2 (or earlier engine) customer-facing line card, the TSC completes packet decapsulation and sends the Layer 2 packet to the customer-facing interface.
  - If the packet is bound to an ISE/E5 customer-facing line card, the TSC sends the packet to the line card for further decapsulation.
- If the backbone-facing line card is ISE/E5, the line card determines if the packet is bound to an Engine 2 (or earlier engine) or an ISE/E5 customer-facing line card.
  - If the packet is bound to an Engine 2 (or earlier engine) customer-facing line card, the packet is sent to the tunnel server card for further decapsulation. Afterward, the decapsulated Layer 2 packet is sent to the Engine 2 (or earlier engine) customer-facing interface.
  - If the packet is bound to an ISE/E5 customer-facing line card, the packet is sent to the ISE/E5 line card for decapsulation.

Note If no tunnel server card is installed, L2TPv3 decapsulation is not supported in the following conditions: - The customer-facing line card is Engine 2 or an earlier engine line card. - The customer-facing line card is ISE/E5 and the backbone-facing line card is non-ISE/5. In these cases, packets received on the backbone-facing interface are dropped. The following warning message is logged: L2TPv3 decapsulation packet dropped.

Cisco 12000 Series Line Cards--General Restrictions

- IS-IS protocol packet fragmentation is supported only for dynamic L2TPv3 sessions.
- Hairpinning is not supported for local-to-local switching. The start and end of an L2TPv3 session must terminate on different routers linked by an IP or MPLS backbone.
- The L2TPv3 feature set is supported as follows. If a tunnel server card is:
  - Installed, and only Engine 2 or older customer-facing line cards are used, normal L2TPv3 tunnel sessions are supported with the L2TPv3 feature set described in the “L2TPv3 Features” topic.
  - Is not installed and ISE/E5 backbone-facing and ISE/E5 customer-facing line cards are used, native L2TPv3 tunnel sessions are supported with the native L2TPv3 feature set described in Table 4.
  - Installed and a combination of Engine 2 or older and ISE/E5 line cards is used as customer-facing line cards, a mixed L2TPv3 tunnel session is supported with the native L2TPv3 feature set described in Table 4.
  - Installed and a ISE/E5 customer-facing and Engine 2 or older backbone-facing line cards are used, a mixed L2TPv3 tunnel session is supported with the native L2TPv3 feature set described in L2TPv3 Encapsulation and L2TPv3 Decapsulation sections above.

- Engine 4 and Engine 4 Plus (E4+) line cards are not supported as the customer-facing line cards in an L2TPv3 tunnel session. However, Engine 4 and Engine 4+ line cards may be used to provide other services in a Layer 2 VPN.
- In a native L2TPv3 tunnel session configured on ISE/E5 interfaces, 802.1q (VLAN) is supported as an L2TPv3 payload starting in Cisco IOS Release 12.0(31)S.

Engine 2 and Earlier Engine-Specific Restrictions

- A dedicated 1-port OC-48c/STM-16c POS/SDH tunnel server card is required for L2TPv3 to function. The server card does not run Engine 2 features.
- TSC-based L2TPv3 tunnel sessions are supported only if a tunnel server card is configured.

To configure the server card, you must enter the **ip unnumbered** command and configure the IP address on the PoS interface of the server card before you configure hardware modules. Then enter the **hw-module slot** *slot-number* **mode server** command.

This initial configuration makes the server card IP-aware for backbones requiring an Address Resolution Protocol (ARP) to be generated by the line card. The backbone types that require this configuration are Ethernet and Spatial Reuse Protocol (SRP).

This configuration is also a requirement for session keepalives. The interface port of the server card is automatically set to loopback internal and no keepalives when the **hw-module slot** *slot-number* **mode server** command is configured.

Note Starting in Cisco IOS Release 12.0(30)S, you must first remove all L2TPv3 xconnect attachment circuits on all Engine-2 or earlier engine customer-facing line cards before you enter the **no hw-module slot** *slot-number* **mode server** command to unconfigure a tunnel server card.

- On the tunnel server card:
  - The IP local interface must be a local loopback interface. Configuring any other interface as the IP local interface results in nonoperational sessions.
  - The IP local interface must be dedicated for the use of L2TPv3 sessions. This interface must not be shared by any other routing or tunneling protocols.
  - The maximum performance of 2.5 million packets per second (pps) is achieved only if you use transmit buffer management (TBM) ASIC ID 60F1. Other ASIC ID versions can cause the performance to be reduced by half. To determine the ASIC value of the line card, use the **execute-on slot** *slot-number* **show controller frfab bma reg | include asic**command, where *slot-number* is the slot number of the server card.
- Cover the optics of the tunnel server card because of possible interference or noise causing cyclic redundancy check (CRC) errors on the line card. These errors are caused by a framer problem in the line card.
- The aggregate performance is bound by the server card limit of 2.5 million pps.
- Because of a framer problem, the server card interfaces accounting in (packets out) are not accurate.
- Only features found in the Vanilla uCode bundle are supported on Engine 2 line cards that are associated with an L2TPv3 session and on a different interface, DLCI, or VLAN of the same line card.

- When you configure an Engine 2 feature, which is not in the Vanilla uCode bundle on an Engine 2 line card, on an interface bound to an L2TPv3 tunnel session, the Vanilla uCode is swapped out. As a result, all traffic through the L2TPv3 session stops on the Engine 2 line card. In this case, you must restore the Vanilla uCode bundle on the line card, and rebind the attachment circuit to the L2TPv3 session as described in the "Configuring the Xconnect Attachment Circuit" topic.
- Configuring output Access Control Lists (ACLs) on any line card swaps out the running Engine 2 line card Vanilla uCode bundle in favor of the ACL uCode bundle. This configuration causes all traffic through the L2TPv3 session to stop on those Engine 2 line cards. If output ACLs are essential on the router, we advise you to originate all L2TPv3 sessions on Engine 0 line cards. Output ACLs do not swap out the server card uCode bundle because of the higher priority.
- Engine 2 line cards do not support Frame Relay switching and Frame Relay L2TPv3 DLCI session on the same line card.
- On Engine 2 line cards, the input Frame Relay permanent virtual circuit (PVC) counters are not updated.
- If the 8-port Fast Ethernet (Engine 1) line card is connected to a hub or switch when L2TPv3 is configured on the ingress side of one or more of its ports, duplicate pckets are generated, causing the router to be flooded with packets. This restriction results from the requirement that CAM filtering is disabled when L2TPv3 is used.
- On the 3-port Gigabyte Ethernet (Engine 2) line card, performance degradation can occur if IP packets coming from a port are sent to the slow path for forwarding. This performance degradation occurs if both the following conditions are met:
  - The port has at least one 802.1q subinterface that is in an L2TPv3 session.
  - The IP packet comes from the port interface itself (not 802.1q encapsulated) or from an 802.1q subinterface that is under the port interface but has no L2TPv3 session bound to it.

Edge Line Card-Specific Restrictions

The following restrictions apply to L2TPv3 sessions configured on IP Services Engine (ISE) and Engine 5 edge line cards:

- Native L2TPv3 sessions are supported only if the feature mode is configured on a customer-facing ISE/E5 line card.

To configure the feature mode, enter the **hw-module slot** *slot-number* **np mode featur**e command. You cannot unconfigure the feature mode on a customer-facing ISE/E5 line card until all L2TPv3 xconnect attachment circuits on the line card are removed.

A backbone-facing ISE/E5 line card can operate in any mode and no special feature mode configuration is required.

- Starting in Cisco IOS Release 12.0(31)S, 802.1q (VLAN) is supported as an L2TPv3 payload in a native L2TPv3 tunnel session configured on ISE/E5 interfaces.
- Native L2TPv3 tunnel sessions on customer-facing ISE/E5 line cards can coexist with tunnel sessions that use a tunnel server card.
- L2TPv3 encapsulation on a customer-facing ISE/E5 line card does not support the L2TPv3 Layer 2 Fragmentation feature.

This means that if you enter the **ip pmtu** command to enable the discovery of a path maximum transmission unit (PMTU) for L2TPv3 traffic, and a customer IP packet exceeds the PMTU, IP fragmentation is not

performed on the IP packet before L2TPv3 encapsulation. These packets are dropped. For more information, see the "L2TPv3 Layer 2 Fragmentation" topic.

The first two tables below show the ISE and E5 interfaces that are supported in a native L2TPv3 tunnel on:

- Customer-facing line cards (ingress encapsulation and egress decapsulation)
- Backbone-facing line cards (ingress decapsulation and egress encapsulation)

Table 2: ISE Interfaces Supported in a Native L2TPv3 Tunnel Session

ISE Line Card	Native L2TPv3 Session on Customer-Facing Interface	Native L2TPv3 Session on Backbone-Facing Interface
4-port OC-3 POS ISE	Supported	Supported
8-port OC-3 POS ISE	Supported	Supported
16-port OC-3 POS ISE	Supported	Supported
4-port OC-12 POS ISE	Supported	Supported
1-port OC-48 POS ISE	Supported	Supported
1-port channelized OC-12 (DS1) ISE	Supported	Not supported
2.5G ISE SPA Interface Processor[1]: • 2-port T3/E3 serial SPA • 4-port T3/E3 serial SPA • 2-port channelized T3 to DS0 SPA • 4-port channelized T3 to DS0 SPA	Supported	Not supported
1-port channelized OC-48 POS ISE	Not supported	Not supported
4-port OC-3 ATM ISE	Supported	Supported
4-port OC-12 ATM ISE	Supported	Supported
4-port Gigabit Ethernet ISE [2]	Supported	Supported

1 For more information about the shared port adapters (SPAs) and SPA interface platforms (SIPs) supported on Cisco 12000 series routers, refer to the Cisco 12000 Series Router SIP and SPA Hardware Installation Guide .

2 The 4-port Gigabit Ethernet ISE line card supports VLAN membership (port-based and VLAN-based) in a native L2TPv3 tunnel session on customer-facing and backbone-facing interfaces. See VLAN for more information.

Table 3: Engine 5 Interfaces Supported in a Native L2TPv3 Tunnel Session

Engine 5 SPA	Native L2TPv3 Session on Customer-Facing Interface	Native L2TPv3 Session on Backbone-Facing Interface
1-port channelized STM-1/OC-3 to DS0	Supported	Not supported
8-port channelized T1/E1	Supported	Not supported
1-port 10-Gigabit Ethernet	Supported	Supported
5-port Gigabit Ethernet	Supported	Supported
10-port Gigabit Ethernet	Not supported	Supported
8-port Fast Ethernet	Supported	Supported
4-port OC-3/STM4 POS	Supported	Not supported
8-port OC-3/STM4 POS	Supported	Not supported
2-port OC-12/STM4 POS	Supported	Not supported
4-port OC-12/STM4 POS	Supported	Not supported
8-port OC-12/STM4 POS	Supported	Not supported
2-port OC-48/STM16 POS/RPR	Not supported	Supported
1-port OC192/STM64 POS/RPR	Not supported	Supported

The table below describes the L2TPv3 features supported in a native L2TPv3 tunnel session and the customer-facing ISE/E5 line cards that support each feature. Note that although native L2TPv3 sessions do not support L2TPv3 Layer 2 (IP packet) fragmentation and slow-path switching features, ATM (as a transport type) and QoS features (traffic policing and shaping) across all media types are supported.

Table 4: L2TPv3 Features Supported in a Native L2TPv3 Session

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
Native L2TPv3 tunneling (fast-path) Supports the same L2TPv3 features that are supported by server card-based L2TPv3 tunneling, except that L2TPv3 Layer 2 (IP packet) fragmentation is not supported. For more information, see the "L2TPv3 Features" section.	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port OC-3 ATM ISE 4-port OC-12 ATM ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 Serial - 4-port T3/E3 Serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 8-port fast Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS
L2TP class and pseudowire class configuration You can create an L2TP template of L2TP control channel parameters that can be inherited by different pseudowire classes configured on a PE router. You can also configure a pseudowire template of L2TPv3 session-level parameters that can be used to configure the transport Layer 2 traffic over an xconnect attachment circuit. For more information, see the sections "Configuring L2TP Control Channel Parameters" and "Configuring the L2TPv3 Pseudowire".	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port OC-3 ATM ISE 4-port OC-12 ATM ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 Serial - 4-port T3/E3 Serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 8-port Fast Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
L2TPv3 tunnel marking and traffic policing on the following types of ingress interfaces, when bound to a native L2TPv3 tunnel session: - 802.1q (VLAN) - ATM - Channelized - Ethernet - Frame Relay DLCIs The following conform, exceed, and violate values for the *action*argument are supported for the **police** command when QoS policies are configured on an ISE/E5 ingress interface bound to a native L2TPv3 tunnel. The **set**commands can also be used to set the IP precedence or DSCP value in the tunnel header of a L2TPv3 tunneled packet on an ingress interface. **conform-action** *actions* : **set-prec-tunnel set-dscp-tunnel transmit** **exceed-action** *actions* : **drop set-clp** (ATM only)**set-dscp-tunnel set-dscp-tunnel** and **set-clp**(ATM only)**set-dscp-tunnel** and **set-frde** (Frame Relay only)**set-frde**(Frame Relay only)**set-prec-tunnel set-prec-tunnel** and **set-clp**(ATM only)**set-prec-tunnel** and **set-frde** (Frame Relay only)**transmit** **violate-action** *actions* : **drop** See " QoS: Tunnel Marking for L2TPv3 Tunnels " for information about how to use the L2TPv3 tunnel marking and traffic policing features on Engine 2 (and earlier engine) interfaces bound to a TSC-based L2TPv3 tunnel session.	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port OC-3 ATM ISE 4-port OC-12 ATM ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 serial - 4-port T3/E3 serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 8-port Fast Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
Frame Relay DLCI-to-DLCI tunneling Frame Relay DLCIs are connected to create an end-to-end Frame Relay PVC. Traffic arriving on a DLCI on one interface is forwarded across an L2TPv3 tunnel to another DLCI on the other interface. For more information, see "DLCI-to-DLCI Switching" in the “Frame Relay” section.	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 serial - 4-port T3/E3 serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS - 2-port OC-48/STM16 POS/RPR
ATM single cell and packed cell relay: VC mode Each VC is mapped to a single L2TPv3 tunnel session. The following ATM cell relay modes are supported: • ATM cells arriving at an ATM interface with the specified VPI and VCI are encapsulated into a single L2TP packet (single cell relay). • ATM cells arriving at an ingress ATM interface are packed into L2TPv3 data packets and transported to the egress ATM interface (packed cell relay). For more information, see the “ATM” section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
ATM single cell and packed cell relay: VP mode ATM cells arriving into a predefined PVP on the ATM interface are transported to a predefined PVP on the egress ATM interface. The following ATM cell relay modes are supported: • A single ATM cell is encapsulated into each L2TPv3 data packet (single cell relay). • Multiple ATM cells are packed into a single L2TPv3 data packet (packed cell relay). For more information, see the "ATM" section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported
ATM single cell relay and packed cell relay: Port mode ATM cells arriving at an ingress ATM interface are encapsulated into L2TPv3 data packets and transported to the egress ATM interface. The following ATM cell relay modes are supported: • A single ATM cell is encapsulated into each L2TPv3 data packet (single cell relay). • Multiple ATM cells are packed into a single L2TPv3 data packet (packed cell relay). For more information, see the "ATM" section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
ATM AAL5 PVC tunneling The ATM AAL5 payload of an AAL5 PVC is mapped to a single L2TPv3 session. For more information, see "ATM AAL5" in the “ATM” section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported
OAM emulation mode for ATM AAL5 OAM local emulation mode for ATM AAL5 payloads is supported. Instead of being passed through the pseudowire, OAM cells are terminated and handled locally. On the L2TPv3-based pseudowire, the CE device sends an SLI message across the pseudowire to notify the peer PE node about the defect, rather than tearing down the session. For more information, s ee "ATM AAL5 over L2TPv3: OAM Local Emulation Mode" in the “ATM” section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported
OAM transparent mode for ATM AAL5 OAM transparent mode for ATM AAL5 payloads is supported. The PE routers pass OAM cells transparently across the L2TPv3 tunnel. For more information, s ee "ATM AAL5 over L2TPv3: OAM Transparent Mode" in the “ATM” section.	4-port OC-3 ATM ISE 4-port OC-12 ATM ISE	Not supported
Ethernet port-to-port tunneling Ethernet frames are tunneled through an L2TP pseudowire. For more information, see the “Ethernet” section.	4-port Gigabit Ethernet ISE	Engine 5 SPAs: - 8-port Fast Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
VLAN-to-VLAN tunneling The following types of VLAN membership are supported in an L2TPv3 tunnel: • Port-based, in which undated Ethernet frames are received • VLAN-based, in which tagged Ethernet frames are received For more information, see the “VLAN” topic.	4-port Gigabit Ethernet ISE	Engine 5 SPAs: - 8-port Fast Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet
Dual rate, 3-Color Marker for traffic policing on Frame Relay DLCIs of ingress interfaces, when bound to a native L2TPv3 tunnel session[3] The dual rate, 3-Color Marker in color-aware and color-blind modes, as defined in RFC 2698 for traffic policing, is supported on ingress ISE interfaces to classify packets. For more information, refer to "QoS: Color-Aware Policer."	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 serial - 4-port T3/E3 serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS - 2-port OC-48/STM16 POS/RPR

Native L2TPv3 Feature	ISE Line Cards (Customer-Facing) Supported	E5 Line Cards (Customer-Facing) Supported
Traffic shaping on ATM and Frame Relay egress interfaces based on class map configuration is supported. Traffic shaping is supported on ATM egress interfaces for the following service categories: • Lowest priority: UBR (unspecified bit rate) • Second priority: VBR-nrt (variable bit rate nonreal-time) • Highest priority: VBR-rt (VBR real time) • Highest priority: CBR (constant bit rate) [4] For more information, see "QoS Traffic Shaping on ATM Line Cards for the Cisco 12000 Series."	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port OC-3 ATM ISE 4-port OC-12 ATM ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port clear channel T3/E3 - 4-port clear channel T3/E3 - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS - 2-port OC-48/STM16 POS/RPR
Layer 2 Virtual Private Network (L2VPN) interworking L2VPN interworking allows attachment circuits using different Layer 2 encapsulation types to be connected over an L2TPv3 pseudowire. On an ISE interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported: ATM AAL5 Ethernet 802.1q (VLAN) Frame Relay DLCI On an Engine 5 interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported: Ethernet 802.1q (VLAN) Frame Relay DLCI	4-port OC-3 POS ISE 8-port OC-3 POS ISE 16-port OC-3 POS ISE 4-port OC-12 POS ISE 1-port OC-48 POS ISE 4-port OC-3 ATM ISE 4-port OC-12 ATM ISE 4-port Gigabit Ethernet ISE 1-port channelized OC-12 (DS1) ISE ISE SPAs: - 2-port T3/E3 serial - 4-port T3/E3 serial - 2-port channelized T3 to DS0 - 4-port channelized T3 to DS0	Engine 5 SPAs: - 1-port channelized STM-1c/OC-3c to DS0 - 8-port channelized T1/E1 - 8-port Fast Ethernet - 8-port 10/100 Ethernet - 1-port 10-Gigabit Ethernet - 2-port Gigabit Ethernet - 5-port Gigabit Ethernet - 10-port Gigabit Ethernet - 4-port OC-3/STM4 POS - 8-port OC-3/STM4 POS - 2-port OC-12/STM4 POS - 4-port OC-12/STM4 POS - 8-port OC-12/STM4 POS - 2-port OC-48/STM16 POS/RPR - 1-port OC192/STM64 POS/RPR

[3] Although the dual-rate, 3-Color Marker policer is not supported on ATM ISE/E5 interfaces, the ATM Forum Traffic Management Version 4.1-compliant Generic Cell Rate Algorithm (GCRA) policer is supported. The GCRA policer uses rate, peak rate, delay tolerance, and ATM maximum burst size, and supports the following options: - set-dscp-tunnel - set-dscp-tunnel and set-clp-transmit - set-prec-tunnel - set-prec-tunnel and set-clp-transmit

[4] Note that VBR-rt and CBR share the same high priority shaping. ATM traffic shaping restricts traffic to the maximum rate configured on an ATM VC or PVP with due priority among the respective service categories. You can configure queue limits for an ATM VC or PVP. The queue limits are dual thresholds in which two different thresholds can be configured for CLP=1 cells and CLP0+1 cells. The CLP1 threshold must be lower than the queue limit threshold so that CLP=1 cells are dropped earlier than CLP=0 cells when packets start to fill the queue.

Frame Relay-Specific Restrictions

- Frame Relay per-DLCI forwarding and port-to-port trunking are mutually exclusive. L2TPv3 does not support the use of both on the same interface at the same time.
- The **xconnect** command is not supported on Frame Relay interfaces directly. For Frame Relay, xconnect is applied under the **connect** command specifying the DLCI to be used.
- Changing the encapsulation type on any interface removes any existing **xconnect** command applied to that interface.
- To use DCE or a Network-to-Network Interface (NNI) on a Frame Relay port, you must configure the **frame-relay switching** command.
- The configuration of an L2TPv3 session on a Multilink Frame Relay (MLFR) bundle interface is supported only on Cisco 12000 series 2-port channelized OC-3/STM-1 (DS1/E1) and 6-port Channelized T3 (T1) line cards. (For more information, see Binding L2TPv3 Sessions to Multilink Frame Relay Interfaces, on page 52.)
- Frame Relay policing is nondistributed on the Cisco 7500 series. By configuring Frame Relay policing, you cause traffic on the affected PVCs to be sent to the RSP for processing.
- Frame Relay support is for 10-bit DLCI addresses. Frame Relay Extended Addressing is not supported.
- Multipoint DLCI is not supported.
- The keepalive is automatically disabled on interfaces that have an xconnect applied to them, except for Frame Relay encapsulation, which is a requirement for LMI.
- Static L2TPv3 sessions do not support Frame Relay LMI interworking.

VLAN-Specific Restrictions

- A PE device is responsible only for static VLAN membership entries that are configured manually on the device. Dynamic VLAN membership entries, entry aging, and membership discovery are not supported.
- Implicit tagging for VLAN memberships operating on other layers, such as membership by MAC address, protocol type at Layer 2, or membership by IP subnet at Layer 3, is not supported.
- Point-to-multipoint and multipoint-to-point configurations are not supported. There is a 1:1 relationship between an attachment circuit and an L2TPv3 session.

ATM VP Mode Single Cell Relay over L2TPv3 Restrictions

- The ATM VP Mode Single Cell Relay over L2TPv3 feature is supported only on the Cisco 7200 and Cisco 7500 series routers with ATM Deluxe PA-A3 interfaces.
- After the ATM VP Mode Single Cell Relay feature is configured for a virtual path connection (VPC), no other permanent virtual circuits (PVCs) are allowed for the same virtual path identifier (VPI).

ATM AAL5 SDU over L2TPv3 and Single Cell Relay VC Mode over L2TPv3 Restrictions

- The ATM AAL5 OAM Emulation over L2TPv3 feature and the ATM Single Cell Relay VC Mode over L2TPv3 feature are supported only on the Cisco 7200, Cisco 7301, Cisco 7304 NSE-100, Cisco 7304 NPE-G100, and Cisco 7500 series routers with ATM Deluxe PA-A3 interfaces.
- Sequencing is supported only for ATM adaptation layer 5 (AAL5) service data unit (SDU) frames or ATM cell relay packets. Sequencing of Operation, Administration, and Maintenance (OAM) cells is not supported.
- Sequencing is supported in CEF mode. If sequencing is enabled with dCEF, all L2TP packets that require sequence number processing are sent to the RSP module.
- L2TPv3 manual mode configuration does not support ATM alarm signaling over the pseudowire.
- The Cisco 7200 series and the Cisco 7500 series ATM driver cannot forward Resource Management (RM) OAM cells over the packet-switched network (PSN) for available bit rate (ABR) ToS. The RM cells are locally terminated.

ATM Port Mode Cell Relay over L2TPv3 Restrictions

- Port mode and virtual path (VP) or VC mode cell relay are mutually exclusive. After the ATM interface is configured for cell relay, no permanent virtual path (PVP) or PVC commands are allowed on that interface.
- ATM port mode cell relay is supported only on the PA-A3-T3, PA-A3-E3, and PA-A3-OC-3 ATM port adapters.
- ATM port mode cell relay is not supported on the PA-A3-8T1IMA and PA-A3-8E1IMA port adapters.

ATM Cell Packing over L2TPv3 Restrictions

- The ATM Cell Packing over L2TPv3 feature is supported only on PA-A3 ATM interfaces on Cisco 7200 and Cisco 7500 routers. Cell packing cannot be configured on other platforms or interface cards.
- A minimum of 2 and a maximum of 28 ATM cells can be packed into an L2TPv3 data packet.

IPv6 Protocol Demultiplexing for L2TPv3 Restrictions

- IPv6 protocol demultiplexing is supported only for Ethernet and terminated DLCI Frame Relay interfaces, PPP traffic, and HDLC traffic.
- IPv6 protocol demultiplexing is supported over noninterworking sessions.
- Frame Relay demultiplexing is supported for point-to-point or multipoint.
- FRF.12 end-to-end fragmentation is supported on the Cisco 7500 and Cisco 12000 series routers only between the CE and PE routers.
- FRF.9 hardware payload compression is supported on the Cisco 7200 series and Cisco 7500 series routers only between the CE and PE routers.
- FRF.9 software payload compression is supported on the Cisco 7500 series routers only between the CE and PE routers.
- FRF.9 process switched payload compression is not supported.
- IETF encapsulation must be used with FRF.9.
- FRF.16 is supported only between the CE and PE routers.
- HDLC restrictions for protocol demultiplexing:
  - IP must be enabled on the interface if you want to configure protocol demultiplexing using the **xconnect** command.
  - IPv6 cannot be enabled on the interface at the same time as the **xconnect** command (with or without protocol demultiplexing).
  - Payload compression is not supported.
- Cisco 12000 series router restrictions for protocol demultiplexing:
  - If a Cisco 12000 series router is acting as the PE with IPv6 protocol demultiplexing using PPP, the remote PE must also be a Cisco 12000 series router.
  - IPv6 protocol demultiplexing for Ethernet encapsulation on Engine-5 line cards is only supported with Version-2 Ethernet SPAs. It is not supported with Version-1 Ethernet SPAs.
  - IPv6 protocol demultiplexing is not supported on the SIP-400 Engine-3 line card.
- IPv6 protocol demultiplexing with PPP encapsulation must be configured in the following order to ensure a working tunnel session:
- Configure the IP address on the interface.
- Enter the encapsulation PPP command.
- Enter the PPP **ipv6cp id proxy** *ipv6-address* command.
- Enter the **xconnect** command with the **match protocol ipv6** command.

If this configuration order is not followed, the tunnel session cannot operate until you issue a **shut/no shut** command on the protocol demultiplexing interface or do an OIR.

L2TPv3 Control Message Hashing Restrictions

- L2TPv3 control channel authentication configured using the **digest** command requires bidirectional configuration on the peer devices. A shared secret must be configured on the communicating nodes.
- For a compatibility matrix of all the L2TPv3 authentication methods, see the Valid Configuration Scenarios table in the IPv6 Protocol Demultiplexing section.

L2TPv3 Digest Secret Graceful Switchover Restrictions

- This feature works only with authentication passwords configured using the L2TPv3 Control Message Hashing feature. L2TPv3 control channel authentication passwords configured with the older, Challenge Handshake Authentication Protocol (CHAP)-like authentication system cannot be updated without tearing down L2TPv3 tunnels and sessions.
- In Cisco IOS Release 12.0(30)S, a maximum of two passwords can be configured simultaneously using the **digest secret** command.

For more information about the L2TPv3 Control Message Hashing feature, see the L2TPv3 Control Message Hashing section.

Quality of Service Restrictions in L2TPv3 Tunneling

Quality of service (QoS) policies configured with the modular QoS CLI (MQC) are supported in L2TPv3 tunnel sessions with the following restrictions:

Frame Relay Interface (Non-ISE/E5)

- On the Cisco 7500 series with distributed CEF (dCEF), in a QoS policy applied to a Frame Relay interface configured for L2TPv3, only the MQC commands **match fr-dlci** in class-map configuration mode and **bandwidth** in policy-map configuration mode are supported. (See the Configuring QoS for L2TPv3 on the Cisco 7500 Series Example task.)
- On the Cisco 12000 series, a QoS policy is supported in TSC-based L2TPv3 tunnel sessions on the Frame Relay interfaces of a 2-port channelized OC-3/STM-1 (DS1/E1) or 6-port channelized T3 (T1) line card with the following restrictions:
  - The **police** command is supported as follows:
    - Only the **transmit** option for the *action* keyword is supported with the **conform-action** command.
    - Only the **set-frde-transmit** option for the *action* keyword is supported with the **exceed-action** command.
    - Only the **drop** option for the *action* keyword is supported with the **violate-action** command.
    - Backward explicit congestion notification (BECN) and forward explicit congestion notification (FECN) configuration are not supported.

- The type of service (ToS) byte must be configured in IP headers of tunneled Frame Relay packets when you configure the L2TPv3 pseudowire (see the Configuring the L2TPv3 Pseudowire task).
- All standard restrictions for configuring QoS on Cisco 12000 series line cards apply to configuring QoS for L2TPv3 on Cisco 12000 series 2-port Channelized OC-3/STM-1 (DS1/E1) or 6-port Channelized T3 line cards.

- On the ingress side of a Cisco 12000 series Frame Relay interface configured for TSC-based L2TPv3 tunneling:
  - Weighted random early detection (WRED) and modified deficit round robin (MDRR) configurations are not supported.

- On the egress side of a Cisco 12000 series Frame Relay interface configured for TSC-based L2TPv3 tunneling:
  - MDRR is the only queueing strategy supported.
  - WRED is the only packet drop strategy supported.
  - MDRR is supported only in the following modes:
    - With both a low latency (priority) queue and class-default queue configured. (The low latency queue is supported only in combination with the class-default queue, and cannot be configured with normal distributed round robin (DRR) queues.)
    - Without a low latency queue configured. (In this case, only six queues are supported, including the class-default queue.)
  - Egress queueing is determined according to the IP precedence values configured for classes of L2TPv3 Frame Relay traffic using the **match ip precedence** command, instead of on a per-DLCI basis.

For an example, see Configuring QoS on a Frame Relay Interface in a TSC-Based L2TPv3 Tunnel Session.

Edge Engine (ISE/E5) Interface

On the Cisco 12000 series, a QoS policy is supported in native L2TPv3 tunnel sessions on ISE/E5 interfaces (see Table 2 and Table 3 for a list of supported line cards) with the following restrictions:

- On a Frame Relay or ATM ISE/E5 interface, traffic policing supports only the following conform, exceed, and violate values for the *action* argument of the **police** command:

conform-action *actions* **set-prec-tunnel set-dscp-tunnel transmit**

exceed-action *actions* **drop set-clp** (ATM only) **set-dscp-tunnel set-dscp-tunnel** and **set-clp** (ATM only) **set-dscp-tunnel** and **set-frde** (Frame Relay only) **set-frde**(Frame Relay only)**set-prec-tunnel set-prec-tunnel** and **set-clp** (ATM only) **set-prec-tunnel** and **set-frde** (Frame Relay only) **transmit**

violate-action *actions* **drop**

- On a Frame Relay ISE/E5 interface:
  - FECN and BECN configuration are not supported.

- Marking the Frame Relay discard eligible (DE) bit using a MQC **set** command is not supported. To set (mark) the DE bit, use the **police exceed-action** *actions* command in policy-map configuration mode.
- Configuring Tofab MDRR or WRED using legacy QoS (not MQC) commands is supported and is based on the tunnel precedence value.
- Egress queueing on a Packet-over-SONET ISE/E5 interface is class-based when configured using MQC.
- Egress queueing on a per-DLCI basis is not supported.

- On an ATM ISE/E5 interface:
  - Traffic shaping is supported on ATM egress interfaces for the following service categories:

- Lowest priority: UBR (unspecified bit rate) Second priority: VBR-nrt (variable bit rate nonreal-time) Highest priority: VBR-rt (VBR real time) Highest priority: CBR (constant bit rate)
- Note that VBR-rt and CBR share the same high-priority shaping. ATM traffic shaping restricts traffic to the maximum rate configured on an ATM VC or PVP with due priority among the respective service categories.
- You can configure queue limits for an ATM VC or PVP. The queue limits are dual thresholds in which two different thresholds can be configured for CLP=1 cells and CLP0+1 cells. The CLP1 threshold must be lower than the queue limit threshold so that CLP=1 cells are dropped earlier than CLP=0 cells when packets start to fill the queue.
  - Although the dual-rate, 3-Color Marker policer is not supported on ATM ISE/E5 interfaces (as on Frame Relay ISE/E5 interfaces), the ATM Forum Traffic Management Version 4.1-compliant Generic Cell Rate Algorithm (GCRA) policer is supported. The GCRA policer uses rate, peak rate, delay tolerance, and ATM maximum burst size, and supports the following actions:

set-dscp-tunnel and **set-clp-transmit**.

Protocol Demultiplexing Interface

Protocol demultiplexing requires a combination of an IP address and the **xconnect** command configured on the interface. The interface is then treated as a regular L3. To apply QoS on the Layer 2 IPv6 traffic, you must classify the IPv6 traffic into a separate class before applying any feature(s) to it.

The following match criterion is used to classify Layer 2 IPv6 traffic on a protocol demultiplexing interface:

```
class-map match-ipv6
     match protocol ipv6
```

In the absence of a class to handle Layer 2 IPv6 traffic, the service policy is not accepted on a protocol demultiplexing interface.

For detailed information about QoS configuration tasks and command syntax, refer to:

- *Cisco IOS Quality of Service Solutions Configuration Guide*
- *Cisco IOS Quality of Service Solutions Command Reference*

Information About Layer 2 Tunneling Protocol Version 3

L2TPv3 provides a method for delivering L2TP services over an IPv4 (non-UDP) backbone network. It encompasses the signaling protocol as well as the packet encapsulation specification.

Migration from UTI to L2TPv3

UTI is a Cisco proprietary protocol that offers a simple high-speed transparent Layer 2-to-Layer 2 service over an IP backbone. The UTI protocol lacks the signaling capability and standards support necessary for large-scale commercial service. To begin to answer the need for a standard way to provide large-scale VPN connectivity over an IP core network, limited migration from UTI to L2TPv3 was introduced in Cisco IOS Release 12.0(21)S. The L2TPv3 feature in Cisco IOS Release 12.0(23)S introduced a more robust version of L2TPv3 to replace UTI.

As described in the section L2TPv3 Header Description, the UTI data header is identical to the L2TPv3 header but with no sequence numbers and an 8-byte cookie. By manually configuring an L2TPv3 session using an 8-byte cookie (see the section, *Manually Configuring L2TPv3 Session Parameters*) and by setting the IP protocol number of outgoing data packets to 120 (as described in the section Configuring the L2TPv3 Pseudowire), you can ensure that a PE running L2TPv3 may interoperate with a peer PE running UTI. However, because UTI does not define a signaling plane, dynamically established L2TPv3 sessions cannot interoperate with UTI.

When a customer upgrades from a pre-L2TPv3 Cisco IOS release to a post-L2TPv3 release, an internal UTI-to-xconnect command-line interface (CLI) migration utility will automatically convert the UTI commands to xconnect and pseudowire class configuration commands without the need for any user intervention. After the CLI migration, the UTI commands that were replaced will not be available. The old-style UTI CLI is hidden from the user.

Note The UTI keepalive feature will *not* be migrated. The UTI keepalive feature will no longer be supported in post-L2TPv3 releases. You should convert to using dynamic L2TPv3 sessions to preserve the functionality provided by the UTI keepalive.

L2TPv3 Operation

L2TPv3 provides similar and enhanced services to replace the current UTI implementation, including the following features:

- Xconnect for Layer 2 tunneling through a pseudowire over an IP network
- Layer 2 VPNs for PE-to-PE router service using xconnect that supports Ethernet, 802.1q (VLAN), Frame Relay, HDLC, and PPP Layer 2 circuits, including both static (UTI-like) and dynamic (using the new L2TPv3 signaling) forwarded sessions

The initial Cisco IOS Release 12.0(23)S features supported only the following features:

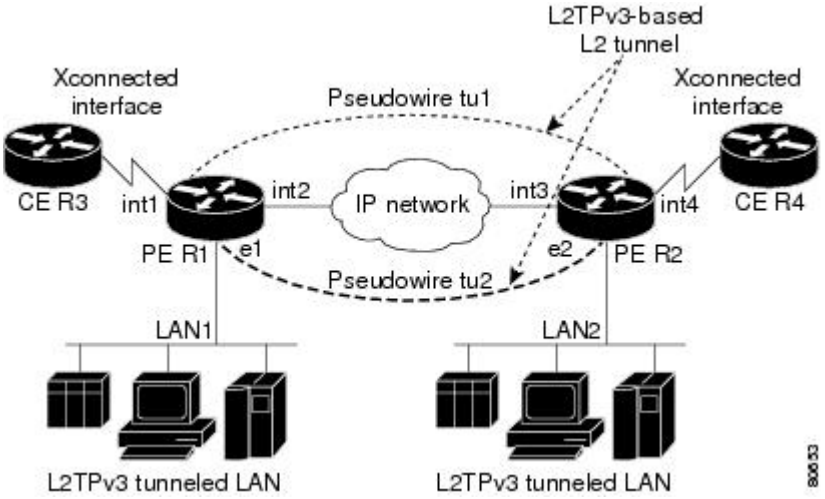
- Layer 2 tunneling (as used in an L2TP access concentrator, or LAC) to an attachment circuit, not Layer 3 tunneling

- L2TPv3 data encapsulation directly over IP (IP protocol number 115), not using User Datagram Protocol (UDP)
- Point-to-point sessions, not point-to-multipoint or multipoint-to-point sessions
- Sessions between the same Layer 2 protocols; for example, Ethernet-to-Ethernet, VLAN-to-VLAN, but not VLAN-to-Ethernet or Frame Relay

The attachment circuit is the physical interface or subinterface attached to the pseudowire.

The figure below shows how the L2TPv3 feature is used for setting up VPNs using Layer 2 tunneling over an IP network. All traffic between two customer network sites is encapsulated in IP packets carrying L2TP data messages and sent across an IP network. The backbone routers of the IP network treat the traffic as any other IP traffic and need not know anything about the customer networks.

Figure 1: L2TPv3 Operation--Example



In the figure above, the PE routers R1 and R2 provide L2TPv3 services. The R1 and R2 routers communicate with each other using a pseudowire over the IP backbone network through a path comprising the interfaces int1 and int2, the IP network, and interfaces int3 and int4.

In this example, the CE routers R3 and R4 communicate through a pair of xconnect Ethernet or VLAN interfaces using an L2TPv3 session. The L2TPv3 session tu1 is a pseudowire configured between interface int1 on R1 and interface int4 on R2. Any packet arriving on interface int1 on R1 is encapsulated and sent through the pseudowire control channel (tu1) to R2. R2 decapsulates the packet and sends it on interface int4 to R4. When R4 needs to send a packet to R3, the packet follows the same path in reverse.

Note the following features regarding L2TPv3 operation:

- All packets received on interface int1 are forwarded to R4. R3 and R4 cannot detect the intervening network.
- For Ethernet interfaces, any packet received from LAN1 by R1 on Ethernet interface e1 are encapsulated directly in IP and sent through the pseudowire session tu2 to R2 interface e2, where it is sent on LAN2.
- A VLAN on an Ethernet interface can be mapped to an L2TPv3 session.

L2TPv3 Benefits

Simplifies Deployment of VPNs

L2TPv3 is an industry-standard Layer 2 tunneling protocol that ensures interoperability among vendors, thus increasing customer flexibility and service availability.

Omits the Need for MPLS

Service providers need not deploy Multiprotocol Label Switching (MPLS) in the core IP backbone to set up VPNs using L2TPv3 over the IP backbone, resulting in operational savings and increased revenue.

Supports Layer 2 Tunneling over IP for Any Payload

L2TPv3 provides enhancements to L2TP to support Layer 2 tunneling of any payload over an IP core network. L2TPv3 defines the base L2TP protocol as being separate from the Layer 2 payload that is tunneled.

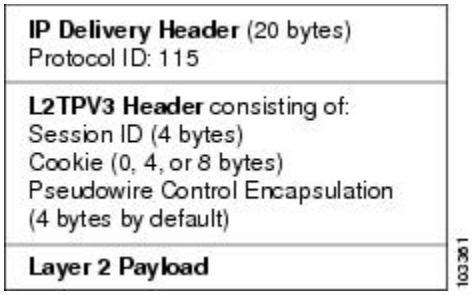
Other Benefits

- Provides cookies for authentication
- Provides session state updates and multiple sessions
- Supports interworking (Ethernet-VLAN, Ethernet-QinQ, and VLAN-QinQ)

L2TPv3 Header Description

The migration from UTI to L2TPv3 also requires the standardization of the UTI header. As a result, the L2TPv3 header has the new format shown in the figure below.

Figure 2: L2TPv3 Header Format



Each L2TPv3 packet contains an L2TPv3 header that includes a unique session ID representing one session and a variable cookie length. The L2TPv3 session ID and the Tunnel Cookie field length are assigned through the CLI. See the section "How to Configure L2TPv3, on page 58" for more information on the CLI commands for L2TPv3.

Session ID

The L2TPv3 session ID is similar to the UTI session ID, and identifies the session context on the decapsulating system. For dynamic sessions, the value of the session ID is selected to optimize the context identification efficiency of the decapsulating system. A decapsulation implementation may therefore elect to support a smaller session ID bit field. In this L2TPv3 implementation, an upper value for the L2TPv3 session ID was set at 023. The L2TPv3 session ID value 0 is reserved for use by the protocol. For static sessions, the session ID is manually configured.

Note The local session ID must be unique on the decapsulating system and is restricted to the least significant ten bits.

Session Cookie

The L2TPv3 header contains a control channel cookie field that is similar to the UTI control channel key field. However, the control channel cookie field has a variable length of 0, 4, or 8 bytes according to the cookie length supported by a given platform for packet decapsulation. The control channel cookie length can be manually configured for static sessions or dynamically determined for dynamic sessions.

The variable cookie length does not present a problem when the same platform is at both ends of an L2TPv3 control channel. However, when different platforms interoperate across an L2TPv3 control channel, both platforms need to encapsulate packets with a 4-byte cookie length.

Pseudowire Control Encapsulation

The L2TPv3 pseudowire control encapsulation consists of 32 bits (4 bytes) and contains information used to sequence L2TP packets and to distinguish AAL5 data and OAM cells for AAL5 SDU mode over L2TPv3. For the purposes of sequencing, only the first bit and bits 8 to 31 are relevant. Bit 1 indicates whether the Sequence Number field, bits 8 to 31, contains a valid sequence number and is to be updated.

L2TPv3 Features

L2TPv3 provides xconnect support for Ethernet, 802.1q (VLAN), Frame Relay, HDLC, and PPP.

Control Channel Parameters

The L2TP class configuration procedure creates a template of L2TP control channel parameters that can be inherited by different pseudowire classes. L2TP control channel parameters are used in control channel authentication, keepalive messages, and control channel negotiation. In an L2TPv3 session, the same L2TP class must be specified in the pseudowire configured on the PE device at each end of the control channel. Configuring L2TP control channel parameters is optional. However, the L2TP class must be configured before it is associated with a pseudowire class (see the Configuring the L2TPv3 Pseudowire task).

L2TPv3 Control Channel Authentication Parameters

Two methods of control channel message authentication are available: the L2TPv3 Control Message Hashing feature and CHAP-style L2TP control channel. The L2TPv3 Control Message Hashing feature introduces a more robust authentication method than the older, CHAP-style L2TP control channel method of authentication. You may choose to enable both the methods of authentication to ensure interoperability with peers that support only one of these methods of authentication, but this configuration will yield control of the authentication method used on the peer PE device. Enabling both the methods of authentication should be considered as an interim solution to solve backward compatibility issues during software upgrades.

The principal difference between the two methods of authentication lies in the L2TPv3 Control Message Hashing feature using the entire message in the hash instead of computing the hash over selected contents of a received control message. In addition, instead of including the hash digest in only the start control channel replay (SCCRP) and start control channel connected (SCCCN) messages, it includes it in all messages.

Support for L2TP control channel authentication is maintained for backward compatibility. Either or both authentication methods can be enabled to allow interoperability with peers supporting only one of the authentication methods.

The table below shows a compatibility matrix for the different L2TPv3 authentication methods. PE1 is running the new authentication method. The possible authentication configurations for PE1 are shown in the first column. The other columns represent PE2 running software with different available authentication options. The tables cells in these columns indicate compatible configuration options for PE2. If any PE1/PE2 authentication configuration poses ambiguity about the authentication method used, the winning authentication method is indicated in bold. If both the old and new authentication methods are enabled on PE1 and PE2, both types of authentication occur.

Table 5: Compatibility Matrix for L2TPv3 Authentication Methods

PE1 Authentication Configuration	PE2 Supporting Old Authentication[5]	PE2 Supporting New Authentication[6]	PE2 Supporting Old and New Authentication[7]
None	None	None New integrity check	None New integrity check
Old authentication	Old authentication	—	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check
New authentication	—	New authentication	New authentication Old authentication and **new authentication**
New integrity check	None	None New integrity check	None New integrity check

PE1 Authentication Configuration	PE2 Supporting Old Authentication[5]	PE2 Supporting New Authentication[6]	PE2 Supporting Old and New Authentication[7]
Old and new authentication	Old authentication	New authentication	Old authentication New authentication **Old and new authentication** **Old authentication** and new integrity check
Old authentication and new integrity check	Old authentication	—	Old authentication **Old authentication** and new authentication **Old authentication** and new integrity check

5 Any PE software that supports only the old CHAP-like authentication system.

6 Any PE software that supports only the new message digest authentication and integrity checking authentication system, but does not understand the old CHAP-like authentication system. This type of software may be implemented by other vendors based on the latest L2TPv3 draft.

7 Any PE software that supports both the old CHAP-like authentication and the new message digest authentication and integrity checking authentication system.

Static L2TPv3 Sessions

Typically, the L2TP control plane is responsible for negotiating session parameters, such as the session ID or the cookie, to set up the session. However, some IP networks require sessions to be configured so that no signaling is required for session establishment. You can set up static L2TPv3 sessions for a PE device by configuring fixed values for the fields in the L2TP data header. A static L2TPv3 session allows the PE device to tunnel Layer 2 traffic as soon as the attachment circuit to which the session is bound comes up.

Static configuration allows sessions to be established without dynamically negotiating control connection parameters. This means that although sessions are displayed in the **show l2tun session** command output, no control channel information is displayed in the **show l2tun tunnel** command output.

Note In an L2TPv3 static session, you can still run the L2TP control channel to perform peer authentication and dead-peer detection. If the L2TP control channel cannot be established or is torn down because of a hello failure, the static session is also torn down.

If you use a static L2TPv3 session, you cannot perform circuit interworking, such as LMI, because there is no facility to exchange control messages. To perform circuit interworking, you must use a dynamic session.

Dynamic L2TPv3 Sessions

A dynamic L2TP session is established through the exchange of control messages containing attribute-value (AV) pairs. Each AV pair contains information about the nature of the Layer 2 link being forwarded, including the payload type and virtual circuit (VC) ID.

Multiple L2TP sessions, one for each forwarded Layer 2 circuit, can exist between a pair of PE devices and can be maintained by a single control channel. Session IDs and cookies are dynamically generated and exchanged as part of a dynamic session setup. Information such as sequencing configuration is also exchanged. Circuit state changes (UP/DOWN) are conveyed using the set link info (SLI) message.

Sequencing

Although the correct sequence of received Layer 2 frames is guaranteed by some Layer 2 technologies (by the nature of the link such as a serial line) or by the protocol itself, forwarded Layer 2 frames may be lost, duplicated, or reordered when they traverse a network as IP packets. If the Layer 2 protocol does not provide an explicit sequencing mechanism, you can configure L2TP to sequence its data packets according to the data channel sequencing mechanism described in the L2TPv3 IETF l2tpext working group draft.

A receiver of L2TP data packets mandates sequencing through the Sequencing Required AV pair when the session is being negotiated. A sender (or one that is manually configured to send sequenced packets) that receives this AV pair uses the Layer 2-specific pseudowire control encapsulation defined in L2TPv3.

You can configure L2TP to drop only out-of-order packets; you cannot configure L2TP to deliver the packets out-of-order. No reordering mechanism is available.

Interworking is not allowed when sequencing is enabled.

Local Switching

Local switching (from one port to another port in the same router) is supported for both static and dynamic sessions. You must configure separate IP addresses for each xconnect statement.

See the Configuration Examples for Layer 2 Tunneling Protocol Version 3 section for an example of how to configure local port switching.

Distributed Switching

Distributed CEF switching is supported for L2TP on the Cisco 7500 series routers.

Note For the Cisco 7500 series, sequencing is supported, but all L2TP packets that require sequence number processing are sent to the RSP.

L2TPv3 Layer 2 Fragmentation

Because the reassembly of fragmented packets is computationally expensive, it is desirable to avoid fragmentation issues in the service provider network. The easiest way to avoid fragmentation issues is to configure the CE routers with an path maximum transmission unit (MTU) value that is smaller than the pseudowire path MTU. However, in scenarios where this is not an option, fragmentation issues must be considered. L2TP initially supported only the following options for packet fragmentation when a packet is determined to exceed the L2TP path MTU:

- Unconditionally drop the packet
- Fragment the packet after L2TP/IP encapsulation

- Drop the packet and send an Internet Control Message Protocol (ICMP) unreachable message back to the CE router

The L2TPv3 Layer 2 Fragmentation feature introduces the ability to allow IP traffic from the CE router to be fragmented before the data enters the pseudowire, forcing the computationally expensive reassembly to occur in the CE network rather than in the service-provider network. The number of fragments that must be generated is determined based on the discovered pseudowire path MTU.

To enable the discovery of the path MTU for Layer 2 traffic, enter the **ip pmtu** command in a pseudowire class configuration (see the Configuring the L2TPv3 Pseudowire section). On the PE router, the original Layer 2 header is then copied to each of the generated fragments, the L2TP/IP encapsulation is added, and the frames are forwarded through the L2TPv3 pseudowire.

Because the Don't Fragment (DF) bit in the Layer 2 encapsulation header is copied from the inner IP header to the encapsulation header, fragmentation of IP packets is performed on any packets received from the CE network that have a DF bit set to 0 and that exceed the L2TP path MTU in size. To prevent the reassembly of fragmented packets on the decapsulation router, you can enter the **ip dfbit set** command in the pseudowire class configuration to enable the DF bit in the outer Layer 2 header.

L2TPv3 Type of Service Marking

When Layer 2 traffic is tunneled across an IP network, information contained in the Type of Service (ToS) bits may be transferred to the L2TP-encapsulated IP packets in one of the following ways:

- If the tunneled Layer 2 frames themselves encapsulate IP packets, it may be desirable to simply copy the ToS bytes of the inner IP packets to the outer IP packet headers. This action is known as "ToS byte reflection."
- You can specify the ToS byte value used by all packets sent across the pseudowire. This is known as "Static ToS byte configuration".

For more details on how to configure ToS, see the Example: Configuring a Negotiated L2TPv3 Session for Local HDLC Switching section.

Keepalive

The keepalive mechanism for L2TPv3 extends only to the endpoints of the tunneling protocol. L2TP has a reliable control message delivery mechanism that serves as the basis for the keepalive mechanism. The keepalive mechanism consists of an exchange of L2TP hello messages.

If a keepalive mechanism is required, the control plane is used, although it may not be used to bring up sessions. You can configure sessions manually.

In the case of static L2TPv3 sessions, a control channel between the two L2TP peers is negotiated through the exchange of start control channel request (SCCRQ), SCCRP, and SCCCN control messages. The control channel is responsible for maintaining only the keepalive mechanism through the exchange of hello messages.

The interval between hello messages is configurable per control channel. If one peer detects that the other peer has gone down through the keepalive mechanism, it sends a StopCCN control message and then notifies all the pseudowires to the peer about the event. This notification results in the teardown of both manually configured and dynamic sessions.

MTU Handling

It is important that you configure a Maximum Transmission Unit (MTU) appropriate for each L2TPv3 tunneled link. The configured MTU size ensures the following:

- The lengths of the tunneled Layer 2 frames fall below the MTU of the destination attachment circuit.
- The tunneled packets are not fragmented, which forces the receiving PE to reassemble them.

L2TPv3 handles the MTU as follows:

- The default behavior is to fragment packets that are larger than the session MTU.
- If you enable the **ip dfbit set** command in the pseudowire class, the default MTU behavior changes so that any packets that cannot fit within the tunnel MTU are dropped.
- If you enable the **ip pmtu** command in the pseudowire class, the L2TPv3 control channel participates in the path MTU (PMTU) discovery.

If you enable this feature, the following processing is performed:

- Internet Control Message Protocol (ICMP) unreachable messages sent back to the L2TPv3 device are deciphered and the tunnel MTU is updated accordingly. To receive ICMP unreachable messages for fragmentation errors, the Don't Fragment (DF) bit in the tunnel header is either set according to the DF bit value received from the CE device or set statically if the **ip dfbit set** option is enabled. The tunnel MTU is periodically reset to the default value based on a periodic timer.
- ICMP unreachable messages are sent back to the clients on the CE side. ICMP unreachable messages are sent to the CE whenever IP packets arrive on the CE-PE interface and have a packet size greater than the tunnel MTU. A Layer 2 header calculation is performed before the ICMP unreachable message is sent to the CE.

L2TPv3 Control Message Hashing

The L2TPv3 Control Message Hashing feature introduces a new and more secure authentication system that replaces the CHAP-like authentication system inherited from L2TPv2, which uses the Challenge and Challenge Response AV pairs in the SCCRQ, SCCRP, and SCCCN messages. The L2TPv3 Control Message Hashing feature incorporates an optional authentication or integrity check for all control messages.

The per-message authentication introduced by the L2TPv3 Control Message Hashing feature is designed to:

- Perform a mutual authentication between L2TP nodes.
- Check integrity of all control messages.
- Guard against control message spoofing and replay attacks that would otherwise be trivial to mount against the network.

The new authentication method uses the following:

- A computed, one-way hash over the header and body of the L2TP control message
- A preconfigured, shared secret that must be defined on the communicating L2TP nodes
- A local and remote random value exchanged using the Nonce AV pairs

Received control messages that lack any of the required security elements are dropped.

L2TPv3 control message integrity checking is a unidirectional mechanism that does not require the configuration of a shared secret. If integrity checking is enabled on the local PE device, control messages are sent with the message digest calculated without the shared secret or Nonce AV pairs and are verified by the remote PE device. If verification fails, the remote PE device drops the control message.

Enabling the L2TPv3 Control Message Hashing feature will impact performance during control channel and session establishment because additional digest calculation of the full message content is required for each sent and received control message. This is an expected trade-off for the additional security provided by this feature. In addition, network congestion may occur if the receive window size is too small. If the L2TPv3 Control Message Hashing feature is enabled, message digest validation must be enabled. Message digest validation deactivates the data path received sequence number update and restricts the minimum local receive window size to 35.

You may choose to configure control channel authentication or control message integrity checking. Control channel authentication requires participation by both peers and a shared secret must be configured on both devices. Control message integrity check is unidirectional and requires configuration on only one of the peers.

L2TPv3 Control Message Rate Limiting

The L2TPv3 Control Message Rate Limiting feature was introduced to counter the possibility of a denial-of-service (DoS) attack on a device running L2TPv3. The L2TPv3 Control Message Rate Limiting feature limits the rate at which SCCRQ control packets arriving at the PE that terminates the L2TPv3 tunnel can be processed. SCCRQ control packets initiate the process of bringing up the L2TPv3 tunnel and require a large amount of control plane resources of the PE device.

No configuration is required for the L2TPv3 Control Message Rate Limiting feature. This feature automatically runs in the background in supported releases.

L2TPv3 Digest Secret Graceful Switchover

Authentication of L2TPv3 control channel messages occurs using a password that is configured on all participating peer PE devices. Before the introduction of this feature, changing this password required removing of the old password from the configuration before adding the new password, causing an interruption in L2TPv3 services. The authentication password must be updated on all peer PE devices, which are often at different physical locations. It is difficult for all peer PE devices to be updated with the new password simultaneously to minimize interruptions in L2TPv3 services.

The L2TPv3 Digest Secret Graceful Switchover feature allows the password used to authenticate L2TPv3 control channel messages to be changed without tearing down the established L2TPv3 tunnels. This feature works only for authentication passwords configured with the L2TPv3 Control Message Hashing feature. Authentication passwords configured with the older, CHAP-like authentication system cannot be updated without tearing down L2TPv3 tunnels.

The L2TPv3 Digest Secret Graceful Switchover feature allows two control channel passwords to be configured simultaneously, so a new control channel password can be enabled without first removing the old password. Established tunnels are rapidly updated with the new password, but continue to use the old password until it is removed from the configuration. This allows authentication to continue normally with peer PE devices that have not yet been updated to use the new password. After all peer PE devices are configured with the new password, the old password can be removed from the configuration.

During the period when both a new and an old password are configured, authentication will occur only with the new password if the attempt to authenticate using the old password fails.

L2TPv3 Pseudowire

The pseudowire class configuration procedure creates a configuration template for the pseudowire. Use this template or class to configure session-level parameters for L2TPv3 sessions that are used to transport attachment circuit traffic over the pseudowire.

The pseudowire configuration specifies the characteristics of the L2TPv3 signaling mechanism, including the data encapsulation type, the control protocol, sequencing, Layer 3 fragmentation, payload-specific options, and IP properties. The setting that determines whether signaling is used to set up the pseudowire is also included.

If you specify the **encapsulation l2tpv3** command, you cannot remove it by using the **no encapsulation l2tpv3** command. You also cannot change the command setting by using the **encapsulation mpls** command. These methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the command, you must delete the pseudowire by using the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire by using the **no pseudowire-class** command, reestablish the pseudowire, and specify the new encapsulation type.

Manual Clearing of L2TPv3 Tunnels

This feature lets you clear L2TPv3 tunnels manually. Before the introduction of this feature, there was no provision to clear a specific L2TPv3 tunnel manually. This functionality provides users more control over an L2TPv3 network.

L2TPv3 Tunnel Management

New and enhanced commands have been introduced to facilitate the management and diagnosis of problems with xconnect configurations. No specific configuration tasks are associated with these commands.

- **debug vpdn**--The output of this command includes authentication failure messages.
- **show l2tun session**--The **hostname** keyword allows the peer hostname to be displayed in the output.
- **show l2tun tunnel**--The **authentication** keyword allows the display of global information about L2TP control channel authentication AV pairs.
- **show xconnect**--The output of this command displays information about xconnect attachment circuits and pseudowires. This command also provides a sortable, single point of reference for information about all xconnect configurations.
- **xconnect logging pseudowire status**--This command enables syslog reporting of pseudowire status events.

For information about these Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the Cisco IOS Master Commands List, All Releases.

Control Message Statistics and Conditional Debugging Command Enhancements

This feature introduces new commands and modifies existing commands for managing control message statistics and conditionally filtering xconnect debug messages.

For this feature, the following commands were introduced:

- **clear l2tun counters** --Clears session counters for Layer 2 tunnels.
- **clear l2tun counters tunnel l2tp** --Clears global or per-tunnel control message statistics.
- **debug condition xconnect** --Allows the conditional filtering of debug messages related to xconnect configurations (allows pseudowire conditional debugging)
- **monitor l2tun counters tunnel l2tp** --Enables or disables the collection of per-tunnel control message statistics.
- **show l2tun counters tunnel l2tp** --Displays global or per-tunnel control message statistics.

For this feature, the following command was modified:

- **show l2tun tunnel** --The **authentication** keyword was removed. The statistics previously displayed by the **show l2tun tunnel authentication** command are now displayed by the **show l2tun counters tunnel l2tp authentication**command.

L2TPv3 Protocol Demultiplexing

The L2TPv3 Protocol Demultiplexing feature introduces the ability to provide native IPv6 support by utilizing a specialized IPv6 network to offload IPv6 traffic from the IPv4 network. The IPv6 traffic is tunneled to the IPv6 network transparently by using L2TPv3 pseudowires without affecting the configuration of the CE devices. IPv4 traffic is routed as usual within the IPv4 network, maintaining the existing performance and reliability of the IPv4 network.

The IPv4 PE devices must be configured to demultiplex the incoming IPv6 traffic from IPv4 traffic. The PE devices facing the IPv6 network do not require the IPv6 configuration. The configuration of the IPv6 network is beyond the scope of this document. For more information on configuring an IPv6 network, see the IPv6 Configuration Guide.

Color Aware Policer on Ethernet over L2TPv3

The Color-Aware Policer enables a "color-aware" method of traffic policing. This feature allows you to police traffic according to the color classification of a packet. The packet color classification is based on packet matching criteria defined for two user-specified traffic classes--the conform-color class and the exceed-color class. These two traffic classes are created using the conform-color command and the metering rates are defined using the police command.

Site of Origin for Border Gateway Protocol VPNs

Site of Origin (SoO) for Border Gateway Protocol Virtual Private Networks (BGP-VPNs) is supported in Cisco IOS Release 12.0(33)S. Site of Origin (SoO) is a concept in a distributed VPN architecture that prevents routing loops in a site which is multi-homed to the VPN backbone and uses AS-OVERRIDE. The mechanism works by applying the SoO tag at the VPN entry point, the provider's edge (PE) equipment. When SoO is enabled, the PE only forwards prefixes to the customer premises equipment (CPE) when the SoO tag of the prefix does not match the SoO tag configured for the CPE.

Each site should be assigned a unique ID value, which is used as the second half of the SoO tag. These ID values used can be repeated for different customers, but not for the same customer. A "site" is considered SoO enabled if it has two or more CPEs that are connected to different PEs and includes at least one non-PE link between them.

SoO is a BGP extended community attribute used to identify when a prefix that originated from a customer site is re-advertised back into that site from a backdoor link. The following format can be used to address the SoO extended community:

<Customer-AS>:<Site-ID>

SoO can now be configured either using inbound route-maps or using the per-neighbor **neighbor soo** command. The SoO value set through the **neighbor soo** command should override the legacy inbound route-map settings when both are configured at the same time.

L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations feature lets you configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with the QinQ or Dot1Q encapsulation. You can set the custom Ethertype to 0x9100, 0x9200, or 0x88A8. This allows interoperability in a multivendor Gigabit Ethernet environment.

L2TPv3 and UTI Feature Comparison

The table below compares L2TPv3 and UTI feature support for the Cisco 7200 and Cisco 7500 series routers.

Table 6: Comparison of L2TPv3 and UTI Feature Support

Feature	L2TPv3	UTI
Maximum number of sessions	Cisco 7200 and Cisco 7500 series:3000	Cisco 7200 and Cisco 7500 series: 1000
Tunnel cookie length	0-, 4-, or 8-byte cookies are supported for the Cisco 7200 series and the Cisco 7500 series routers.	8 bytes
Static sessions	Supported in Cisco IOS Release 12.0(21)S.	Supported
Dynamic sessions	Supported in Cisco IOS Release 12.0(23)S.	Not supported
Static ToS	Supported in Cisco IOS Release 12.0(23)S.	Supported
MQC ToS	Supported in Cisco IOS Release 12.0(27)S.	Supported
Inner IP ToS mapping	Supported on the Cisco 7200 series routers and Cisco 7500 series routers.	Not supported
802.1p mapping	Not supported.	Not supported

Feature	L2TPv3	UTI
Keepalive	Supported in Cisco IOS Release 12.0(23)S.	Not supported
Path MTU discovery	Supported on the Cisco 7200 series and Cisco 7500 series routers.	Not supported
ICMP unreachable	Supported on the Cisco 7200 series and Cisco 7500 series routers.	Not supported
VLAN rewrite	Supported on the Cisco 7200 series and Cisco 7500 series routers in Cisco IOS Release 12.0(23)S.	Supported
VLAN and non-VLAN translation	To be supported in a future release.	Not supported
Port trunking	Supported in Cisco IOS Release 12.0(23)S.	Supported
IS-IS packet fragmentation through an L2TPv3 session	Supported on the Cisco 7200 series and Cisco 7500 series routers, and on the Cisco 10720 Internet router in Cisco IOS Release 12.0(24)S.	Not supported
L2TPv3 Layer 2 (IP packet) fragmentation through an L2TPv3 session	Supported on the Cisco 7200 series and Cisco 7500 series routers in Cisco IOS Release 12.0(24)S. Supported on the Cisco 10720 Internet router in Cisco IOS Release 12.0(32)SY.	Not supported
Payload sequence number checking	Supported on the Cisco 7500 series in Cisco IOS Release 12.0(28)S.	Not supported
MIB support	IfTable MIB for the attachment circuit.	IfTable MIB for the session interface.

Supported L2TPv3 Payloads

Note Each L2TPv3 tunneled packet includes the entire Layer 2 frame of the payloads described in this section. If sequencing is required (see the Sequencing section), a Layer 2-specific sublayer (see the Pseudowire Control Encapsulation section) is included in the L2TPv3 header to provide the Sequence Number field.

Frame Relay

Port-to-Port Trunking

Port-to-port trunking is where two CE Frame Relay interfaces are connected as by a leased line (UTI raw mode). All traffic arriving on one interface is forwarded transparently across the pseudowire to the other interface.

For example, in Port-to-Port Trunking, if the two CE routers are connected by a virtual leased line, the PE routers transparently transport all packets between CE R3 and CE R4 over a pseudowire. PE R1 and PE R2 do not examine or change the DLCIs, and do not participate in the LMI protocol. The two CE routers are LMI peers. There is nothing Frame Relay-specific about this service as far as the PE routers are concerned. The CE routers should be able to use any encapsulation based on HDLC framing without needing to change the provider configuration.

DLCI-to-DLCI Switching

Frame Relay DLCI-to-DLCI switching is where individual Frame Relay DLCIs are connected to create an end-to-end Frame Relay PVC. Traffic arriving on a DLCI on one interface is forwarded across the pseudowire to another DLCI on the other interface.

For example, in DLCI-to-DLCI Switching, CE R3 and PE R1 are Frame Relay LMI peers; CE R4 and PE R2 are also LMI peers. You can use a different type of LMI between CE R3 and PE R1 compared to what you use between CE R4 and PE R2.

The CE devices may be a Frame Relay switch or end-user device. 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. Note that, in DLCI-to-DLCI Switching, two Frame Relay PVC segments are connected by a pseudowire. Frame Relay header flags (FECN, BECN, C/R, DE) are preserved across the pseudowire.

PVC Status Signaling

PVC status signaling is propagated toward Frame Relay end users by the LMI protocol. You can configure the LMI to operate in any of the following modes:

- UNI DTE mode--PVC status is not reported, only received.
- UNI DCE mode--PVC status is reported but not received.
- NNI mode--PVC status is reported and received independently.

L2TPv3 supports all three modes.

The PVC status should be reported as ACTIVE only if the PVC is available from the reporting device to the Frame Relay end-user device. All interfaces, line protocols, and pseudowires must be operational between the reporting device and the Frame Relay end-user device.

Note that any keepalive functions on the session are independent of Frame Relay, but any state changes that are detected are fed into the PVC status reporting. For example, the L2TP control channel uses hello packets as a keepalive function. If the L2TPv3 keepalive fails, all L2TPv3 sessions are torn down. Loss of the session is notified to Frame Relay, which can then report PVCs INACTIVE to the CE devices.

For example, in PVC Status Signaling, CE R3 reports ACTIVE to PE R1 only if the PVC is available within CE R3. When CE R3 is a switch, it reports all the way to the user device in the customer network.

PE R1 reports ACTIVE to CE R3 only if the PVC is available within PE R1 and all the way to the end-user device (through PE R2 and CE R3) in the other customer VPN site.

The ACTIVE state is propagated hop-by-hop, independently in each direction, from one end of the Frame Relay network to the other end.

Sequencing

Frame Relay provides an ordered service in which packets sent to the Frame Relay network by one end-user device are delivered in order to the other end-user device. When switching is occurring over the pseudowire, packet ordering must be able to be preserved with a very high probability to closely emulate a traditional Frame Relay service. If the CE router is not using a protocol that can detect misordering itself, configuring sequence number processing may be important. For example, if the Layer 3 protocol is IP and Frame Relay is therefore used only for encapsulation, sequencing is not required. To detect misordering, you can configure sequence number processing separately for transmission or reception. For more information about how to configure sequencing, see the section "Example: Configuring a Negotiated L2TPv3 Session for Local HDLC Switching , on page 98."

ToS Marking

The ToS bytes in the IP header can be statically configured or reflected from the internal IP header. The Frame Relay discard eligible (DE) bit does not influence the ToS bytes.

CIR Guarantees

To provide committed information rate (CIR) guarantees, you can configure a queueing policy that provides bandwidth to each DLCI to the interface facing the customer network on the egress PE.

Note CIR guarantees are supported only on the Cisco 7500 series with dCEF. This support requires that the core has sufficient bandwidth to handle all CE traffic and that the congestion occurs only at the egress PE.

Binding L2TPv3 Sessions to Multilink Frame Relay Interfaces

The configuration of an L2TPv3 session on a Multilink Frame Relay (MLFR) bundle interface is supported only on Cisco 12000 series 2-port channelized OC-3/STM-1 (DS1/E1) and 6-port channelized T3 (T1) line cards.

The Multilink Frame Relay feature introduces functionality based on the Frame Relay Forum Multilink Frame Relay UNI/NNI Implementation Agreement (FRF.16). This feature provides a cost-effective way to increase bandwidth for particular applications by enabling multiple serial links to be aggregated into a single bundle of bandwidth.

For an example of how to configure L2TPv3 tunneling on a multilink Frame Relay bundle interface, see Configuring MLFR for L2TPv3 on the Cisco 12000 Series Example, on page 114.

For information about how configure and use the MLFR feature, refer to the Multilink Frame Relay (FRF.16) publication .

Ethernet

An Ethernet frame arriving at a PE device is simply encapsulated in its entirety with an L2TP data header. At the other end, a received L2TP data packet is stripped of its L2TP data header. The payload, an Ethernet frame, is then forwarded to the appropriate attachment circuit.

Because the L2TPv3 tunneling protocol serves essentially as a bridge, it need not examine any part of an Ethernet frame. Any Ethernet frame received on an interface is tunneled, and any L2TP-tunneled Ethernet frame is forwarded out of the interface.

Note Because of the way in which L2TPv3 handles Ethernet frames, an Ethernet interface must be configured to promiscuous mode to capture all traffic received on the Ethernet segment attached to the device. All frames are tunneled through the L2TP pseudowire.

VLAN

L2TPv3 supports VLAN memberships in the following ways:

- Port-based, in which undated Ethernet frames are received
- VLAN-based, in which tagged Ethernet frames are received

In L2TPv3, Ethernet xconnect supports port-based VLAN membership and the reception of tagged Ethernet frames. A tagged Ethernet frame contains a tag header (defined in 802.1Q), which is 4 bytes long and consists of a 2-byte tag protocol identifier (TPID) field and a 2-byte tag control information (TCI) field. The TPID indicates that a TCI follows. The TCI is further broken down into the following three fields:

- User priority field
- Canonical format indicator (CFI)
- A 12-bit VLAN ID (VID)

For L2TPv3, an Ethernet subinterface configured to support VLAN switching may be bound to an xconnect service so that all Ethernet traffic, tagged with a VID specified on the subinterface, is tunneled to another PE. The VLAN Ethernet frames are forwarded in their entirety. The receiving PE may rewrite the VID of the tunneled traffic to another value before forwarding the traffic onto an attachment circuit.

To successfully rewrite VLANs, it may be necessary to disable the Spanning Tree Protocol (STP). This can be done on a per-VLAN basis by using the **no spanning-tree vlan** command.

Note Because of the way in which L2TPv3 handles VLAN packets, the Ethernet interface must be configured in promiscuous mode to capture all traffic received on the Ethernet segment attached to the device. All frames are tunneled through the L2TP pseudowire.

HDLC

L2TPv3 encapsulates an HDLC frame arriving at a PE in its entirety (including the Address, Control, and Protocol fields, but not the Flag fields and the frame check sequence) with an L2TP data header.

PPP

PEs that support L2TPv3 forward PPP traffic using a "transparent pass-through" model, in which the PEs play no role in the negotiation and maintenance of the PPP link. L2TPv3 encapsulates a PPP frame arriving at a PE in its entirety (including the HDLC Address and Control fields) with an L2TP data header.

ATM

L2TPv3 can connect two isolated ATM clouds over a packet-switched network (PSN) while maintaining an end-to-end ATM Service Level Agreement (SLA). The ATM Single Cell Relay features forward one ATM cell per packet. The ATM Cell Packing over L2TPv3 features allows multiple ATM frames to be packed into a single L2TPv3 data packet. All packets are transparently forwarded over the L2TPv3 pseudowire.

Note VPI or VPI/VCI rewrite is not supported for any ATM transport mode. Both pairs of PE to CE peer routers must be configured with matching VPI or VCI values except in OAM local emulation mode. For example, if PE1 and CE1 are connected by PVC 10/100, PE2 and CE2 should also be connected by PVC 10/100.

The table below shows the releases that introduced support for the ATM cell relay features.

Table 7: Release Support for the ATM Cell Relay Features

Transport Type	Single Cell Relay	Packed Cell Relay
VC mode	12.0(28)S, 12.2(25)S	12.0(29)S
VP mode	12.0(25)S, 12.2(25)S	12.0(29)S
Port mode	12.0(29)S, 12.2(25)S4	12.0(29)S

ATM Single Cell Relay VC Mode over L2TPv3

The ATM Single Cell Relay VC mode over L2TPv3 feature maps one VC to a single L2TPv3 session. All ATM cells arriving at an ATM interface with the specified VPI and VCI are encapsulated into a single L2TP packet. Each ATM cell will have a 4-byte ATM cell header without Header Error Control Checksum (HEC) and a 48-byte ATM cell payload.

The ATM Single Cell Relay VC mode feature can be used to carry any type of AAL traffic over the pseudowire. It will not distinguish OAM cells from User data cells. In this mode, Performance and Security OAM cells are also transported over the pseudowire.

ATM VP Mode Single Cell Relay over L2TPv3

The ATM VP Mode Single Cell Relay over L2TPv3 feature allows cells coming into a predefined PVP on the ATM interface to be transported over an L2TPv3 pseudowire to a predefined PVP on the egress ATM interface. A single ATM cell is encapsulated into each L2TPv3 data packet.

ATM Port Mode Cell Relay over L2TPv3

The ATM Port Mode Cell Relay over L2TPv3 feature packs ATM cells arriving at an ingress ATM interface into L2TPv3 data packets and transports them to the egress ATM interface. A single ATM cell is encapsulated into each L2TPv3 data packet.

ATM Cell Packing over L2TPv3

The ATM Cell Packing over L2TPv3 feature enhances throughput and uses bandwidth more efficiently than the ATM cell relay features. Instead of a single ATM cell being packed into each L2TPv3 data packet, multiple ATM cells can be packed into a single L2TPv3 data packet. ATM cell packing is supported for Port mode, VP mode, and VC mode. Cell packing must be configured on the PE devices. No configuration is required on the CE devices.

ATM AAL5 over L2TPv3

The ATM AAL5 over L2TPv3 feature maps the AAL5 payload of an AAL5 PVC to a single L2TPv3 session. This service will transport OAM and RM cells, but does not attempt to maintain the relative order of these cells with respect to the cells that comprise the AAL5 common part convergence sublayer protocol data unit (CPCS-PDU). OAM cells that arrive during the reassembly of a single AAL5 CPCS-PDU are sent immediately over the pseudowire, followed by the AAL5 payload without the AAL5 pad and trailer bytes.

VC Class Provisioning for L2TPv3

Beginning in Cisco IOS Release 12.0(30)S, ATM AAL5 encapsulation over L2TPv3 can be configured in VC class configuration mode in addition to ATM VC configuration mode. The ability to configure ATM encapsulation parameters in VC class configuration mode provides greater control and flexibility for AAL5 encapsulation configurations.

OAM Transparent Mode

In OAM transparent mode, the PEs will pass the following OAM cells transparently across the pseudowire:

- F5 segment and end-to-end Fault Management (FM) OAM cells
- RM OAM cells, except Performance Management (PM) and Security OAM cells

Note The Cisco 7200 and the Cisco 7500 ATM driver cannot forward RM cells over the PSN for ABR ToS. The RM cells are locally terminated.

VPI or VPI/VCI rewrite is not supported for any ATM transport mode. Both pairs of PE to CE peer routers must be configured with matching VPI and VCI values except in OAM local emulation mode. For example, if PE1 and CE1 are connected by PVC 10/100, PE2 and CE2 should also be connected by PVC 10/100.

OAM Local Emulation Mode

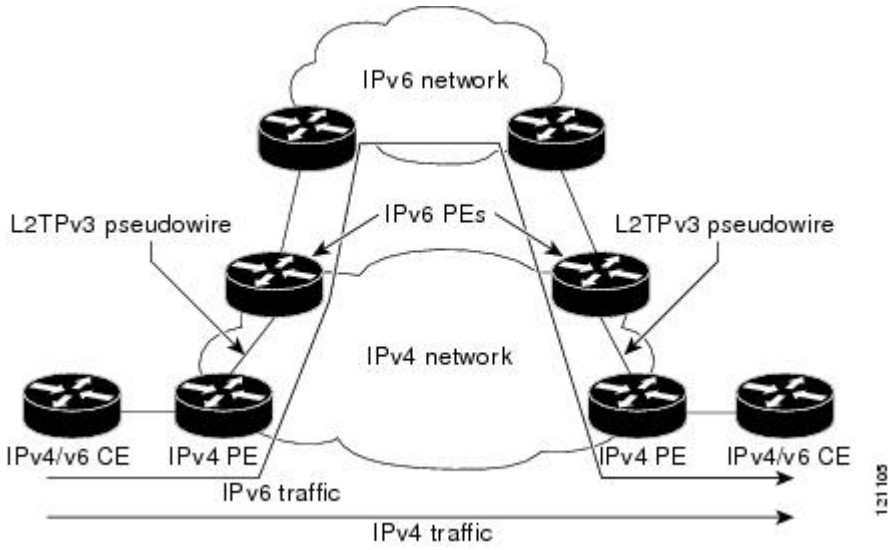
In OAM Local Emulation mode, OAM cells are not passed through the pseudowire. All F5 OAM cells are terminated and handled locally. On the L2TPv3-based pseudowire, the CE device sends an SLI message across the pseudowire to notify the peer PE node about the defect, rather than tearing down the session. The defect can occur at any point in the link between the local CE and the PE. OAM management can also be enabled on the PE node using existing OAM management configurations.

IPv6 Protocol Demultiplexing

Upgrading a service provider network to support IPv6 is a long and expensive process. As an interim solution, the Protocol Demultiplexing for L2TPv3 feature introduces the ability to provide native IPv6 support by setting up a specialized IPv6 network and offloading IPv6 traffic from the IPv4 network. IPv6 traffic is tunneled transparently to the IPv6 network using L2TPv3 pseudowires without affecting the configuration of the CE devices. IPv4 traffic is routed as usual within the IPv4 network, maintaining the existing performance and reliability of the IPv4 network.

The figure below shows a network deployment that offloads IPv6 traffic from the IPv4 network to a specialized IPv6 network. The PE devices demultiplex the IPv6 traffic from the IPv4 traffic. IPv6 traffic is routed to the IPv6 network over an L2TPv3 pseudowire, while IPv4 traffic is routed normally. The IPv4 PE devices must be configured to demultiplex the incoming IPv6 traffic from the IPv4 traffic. The PE devices facing the IPv6 network do not require the IPv6 configuration.

Figure 3: Protocol Demultiplexing of IPv6 Traffic from IPv4 Traffic



If no IP address is configured, the protocol demultiplexing configuration is rejected. If an IP address is configured, the **xconnect** command configuration is rejected unless protocol demultiplexing is enabled in xconnect configuration mode before exiting that mode. If an IP address is configured with an **xconnect** command configuration and protocol demultiplexing is enabled, the IP address cannot be removed. To change or remove the configured IP address, the **xconnect** command configuration must first be disabled.

The table below shows the valid combinations of configurations.

Table 8: Valid Configuration Scenarios

Scenario	IP Address	xconnect Configuration	Protocol Demultiplexing Configuration
Routing	Yes	No	--
L2VPN	No	Yes	No
IPv6 Protocol Demultiplexing	Yes	Yes	Yes

Supported Port Adapters for the Cisco 7200 Series and Cisco 7500 Series Routers

The following port adapters support L2TPv3 on the Cisco 7200 series and Cisco 7500 series routers:

- Single-port Fast Ethernet 100BASE-TX
- Single-port Fast Ethernet 100BASE-FX
- Dual-port Fast Ethernet 100BASE-TX
- Dual-port Fast Ethernet 100BASE-FX
- Gigabit Ethernet port adapter
- 12-port Ethernet/2-port FE adapter
- 4-port synchronous serial port adapter
- Enhanced 4-port synchronous serial port adapter
- 8-port synchronous serial port adapter
- Single-port HSSI adapter
- Dual-port HSSI adapter
- Single-port enhanced OC-3 ATM port adapter
- 8-port multichannel E1 G.703/G.704 120-ohm interfaces
- 2-port multichannel E1 G.703/G.704 120-ohm interfaces
- 8-port multichannel T1 with integrated data service units (DSUs)
- 8-port multichannel T1 with integrated channel service units (CSUs) and DSUs
- 4-port multichannel T1 with integrated CSUs and DSUs
- 2-port multichannel T1 with integrated CSUs and DSUs
- 8-port multichannel T1/E1
- 1-port multichannel T3 interface
- 1-port multichannel E3 interface

- 2-port enhanced multichannel T3 port adapter
- Single-port T3 port adapter
- Single-port E3 port adapter
- 2-port T3 port adapter
- 2-port T3 port adapter
- Single-port Packet over SONET (PoS), single-mode, long reach
- Single-port PoS, single-mode, intermediate reach
- Single-port PoS, multimode
- Eight-port T1 ATM port adapter with inverse multiplexing over ATM (IMA)
- Eight-port E1 ATM port adapter with IMA

The following port adapters support L2TPv3 on the Cisco 7200 series routers only:

- 8-port Ethernet adapter
- 4-port Ethernet adapter

How to Configure L2TPv3

Configuring L2TP Control Channel Parameters

After you enter L2TP class configuration mode, you can configure L2TP control channel parameters in any order. If you have multiple authentication requirements, you can configure multiple sets of L2TP class control channel parameters with different L2TP class names. However, only one set of parameters can be applied to a connection between any pair of IP addresses.

Configuring L2TP Control Channel Timing Parameters

The following L2TP control channel timing parameters can be configured in L2TP class configuration mode:

- Packet size of the receive window used for the control channel
- Retransmission parameters used for control messages
- Timeout parameters used for the control channel

This task configures a set of timing control channel parameters in an L2TP class. All of the timing control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, the default values are applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **receive-window** *size*
5. **retransmit** {**initial retries** *initial-retries*| **retries** *retries*| **timeout** {**max** | **min**} *timeout*}
6. **timeout setup** *seconds*

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	**l2tp-class** [*l2tp-class-name*] **Example:** `Router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name*for each one.
Step 4	**receive-window** *size* **Example:** `Router(config-l2tp-class)# receive-window 30`	(Optional) Configures the number of packets that can be received by the remote peer before backoff queueing occurs. • The valid values range from 1 to the upper limit the peer has for receiving packets. The default value is the upper limit.
Step 5	**retransmit** {**initial retries** *initial-retries*| **retries** *retries*| **timeout** {**max**	**min**} *timeout*} **Example:** `Router(config-l2tp-class)# retransmit retries 10`	(Optional) Configures parameters that affect the retransmission of control packets. • **initial retries** --specifies how many SCCRQs are re-sent before giving up on the session. Valid values for the *initial-retries* argument range from 1 to 1000. The default value is 2. • **retries** --specifies how many retransmission cycles occur before determining that the peer PE router does not respond. Valid values for the *retries* argument range from 1 to 1000. The default value is 15. • **timeout** {**max**	**min**}--specifies maximum and minimum retransmission intervals (in seconds) for resending control packets. Valid values for

	Command or Action	Purpose
		the *timeout* argument range from 1 to 8. The default maximum interval is 8; the default minimum interval is 1.

Step 6	**timeout setup** *seconds*
	
	Example:
	`Router(config-l2tp-class)#`
	`timeout setup 400`
		(Optional) Configures the amount of time, in seconds, allowed to set up a control channel.
		• Valid values for the *seconds* argument range from 60 to 6000. The default value is 300.

Configuring L2TPv3 Control Channel Authentication Parameters

Configuring Authentication for the L2TP Control Channel

The L2TP control channel method of authentication is the older, CHAP-like authentication system inherited from L2TPv2.

The following L2TP control channel authentication parameters can be configured in L2TP class configuration mode:

- Authentication for the L2TP control channel
- Password used for L2TP control channel authentication
- Local hostname used for authenticating the control channel

This task configures a set of authentication control channel parameters in an L2TP class. All of the authentication control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, default values are applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **authentication**
5. **password** [**0** | **7**] *password*
6. **hostname** *name*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 4	**authentication** **Example:** `Device(config-l2tp-class)# authentication`	(Optional) Enables authentication for the control channel between PE devices.

Step 5	**password** [**0** | **7**] *password*
	
	Example:
	`Device(config-l2tp-class)# password cisco`
		(Optional) Configures the password used for control channel authentication.
		• [**0** | **7**]—(Optional) Specifies the input format of the shared secret. The default value is **0**.
		  • **0**—Specifies that a plain-text secret is entered.
		  • **7**—Specifies that an encrypted secret is entered.
		• *password*—Defines the shared password between peer devices.

Step 6	**hostname** *name* **Example:** `Device(config-l2tp-class)# hostname yb2`	(Optional) Specifies a hostname used to identify the device during L2TP control channel authentication. • If you do not use this command, the default hostname of the device is used.
Step 7	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring L2TPv3 Control Message Hashing

This task configures L2TPv3 Control Message Hashing feature for an L2TP class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}]
5. **digest** check
6. **hidden**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.

Step 4	**digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}]
	
	Example:
	`Device(config-l2tp-class)# digest secret cisco hash sha`
		(Optional) Enables L2TPv3 control channel authentication or integrity checking.
		• **secret—**(Optional) Enables L2TPv3 control channel authentication.
		Note If the **digest** command is issued without the **secret** keyword option, L2TPv3 integrity checking is enabled.
		• [**0** | **7**]**—**Specifies the input format of the shared secret. The default value is **0**.
		  • **0—**Specifies that a plain-text secret is entered.
		  • **7—**Specifies that an encrypted secret is entered.
		• *password***—**Defines the shared secret between peer devices. The value entered for the *password* argument must be in the format that matches the input format specified by the [**0** | **7**] keyword option.

	Command or Action	Purpose
		• **hash** {**md5** | **sha**}—(Optional) Specifies the hash function to be used in per-message digest calculations. • **md5**—Specifies HMAC-MD5 hashing. • **sha**—Specifies HMAC-SHA-1 hashing. The default hash function is **md5**.
Step 5	**digest** check **Example:** `Device(config-l2tp-class)# digest check`	(Optional) Enables the validation of the message digest in received control messages. • Validation of the message digest is enabled by default. **Note** Validation of the message digest cannot be disabled if authentication has been enabled using the **digest secret** command. If authentication has not been configured with the **digest secret** command, the digest check can be disabled to increase performance.
Step 6	**hidden** **Example:** `Device(config-l2tp-class)# hidden`	(Optional) Enables AV pair hiding when sending control messages to an L2TPv3 peer. • AV pair hiding is disabled by default. • Only the hiding of the cookie AV pair is supported. • If a cookie is configured in L2TP class configuration mode (see the section *"Manually Configuring L2TPv3 Session Parameters*"), enabling AV pair hiding causes that cookie to be sent to the peer as a hidden AV pair using the password configured with the **digest secret** command. **Note** AV pair hiding is enabled only if authentication has been enabled using the **digest secret** command, and no other authentication method is configured.
Step 7	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring L2TPv3 Digest Secret Graceful Switchover

Perform this task to make the transition from an old L2TPv3 control channel authentication password to a new L2TPv3 control channel authentication password without disrupting established L2TPv3 tunnels.

Before You Begin

Before performing this task, you must enable control channel authentication as documented in the Configuring L2TPv3 Control Message Hashing task.

Note This task is not compatible with authentication passwords configured with the older, CHAP-like control channel authentication system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** *l2tp-class-name*
4. **digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}]
5. **end**
6. **show l2tun tunnel all**
7. **configure terminal**
8. **l2tp-class** [*l2tp-class-name*]
9. **no digest** [**secret** [**0** | **7**] *password* [**hash** {**md5** | **sha**}]
10. **end**
11. **show l2tun tunnel all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** *l2tp-class-name* **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 4	**digest** [**secret** [**0**	**7**] *password*] [**hash** {**md5**	**sha**}] **Example:** `Device(config-l2tp-class)# digest secret cisco2 hash sha`	Configures a new password to be used in L2TPv3 control channel authentication. • A maximum of two passwords may be configured at any time. **Note** Authentication will now occur using both the old and new passwords.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-l2tp-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 6	**show l2tun tunnel all** **Example:** `Device# show l2tun tunnel all`	(Optional) Displays the current state of Layer 2 tunnels and information about configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and control channel information. • Tunnels should be updated with the new control channel authentication password within a matter of seconds. If a tunnel does not update to show that two secrets are configured after several minutes have passed, the tunnel can be cleared manually and a defect report should be filed with the Cisco Technical Assistance Center (TAC). To clear an L2TPv3 tunnel manually, perform the task described in the section "Manually Clearing L2TPv3 Tunnels. " **Note** Issue this command to determine whether any tunnel is using the new password for control channel authentication. The output displayed for each tunnel in the specified L2TP class should show that two secrets are configured.
Step 7	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 8	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.
Step 9	**no digest** [**secret** [**0** | **7**] *password* [**hash** {**md5** | **sha**}] **Example:** `Device(config-l2tp-class)# no digest secret cisco hash sha`	Removes the old password used in L2TPv3 control channel authentication. **Note** Do not remove the old password until all peer PE devices have been updated with the new password.
Step 10	**end** **Example:** `Device(config-l2tp-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 11	**show l2tun tunnel all** **Example:** `Device# show l2tun tunnel all`	(Optional) Displays the current state of Layer 2 tunnels and information about configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and control channel information. • Tunnels should no longer be using the old control channel authentication password. If a tunnel does not update to show that only one secret is configured after several minutes have passed, that tunnel can be cleared

	Command or Action	Purpose
		manually and a defect report should be filed with TAC. To clear an L2TPv3 tunnel manually, perform the task described in the section "Manually Clearing L2TPv3 Tunnels." **Note** Issue this command to ensure that all tunnels are using only the new password for control channel authentication. The output displayed for each tunnel in the specified L2TP class should show that one secret is configured.

Configuring L2TP Control Channel Maintenance Parameters

The L2TP hello packet keepalive interval control channel maintenance parameter can be configured in L2TP class configuration mode.

This task configures the interval used for hello messages in an L2TP class. This control channel parameter configuration is optional. If this parameter is not configured, the default value is applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **hello** *interval*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Device(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, to configure multiple L2TP classes, you must specify a unique *l2tp-class-name* for each one.

	Command or Action	Purpose
Step 4	**hello** *interval* **Example:** `Device(config-l2tp-class)# hello 100`	(Optional) Specifies the exchange interval (in seconds) used between L2TP hello packets. • Valid values for the *interval* argument range from 0 to 1000. The default value is 60.
Step 5	**exit** **Example:** `Device(config-l2tp-class)# exit`	Exits L2TP class configuration mode.

Configuring the L2TPv3 Pseudowire

Perform this task to configure the L2TPv3 pseudowire.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation l2tpv3**
5. **protocol** {**l2tpv3** | **none**} [*l2tp-class-name*]
6. **ip local interface** *interface-name*
7. **ip pmtu**
8. **ip tos** {**value** *value* | **reflect**}
9. **ip dfbit set**
10. **ip ttl** *value*
11. **ip protocol** {**l2tp** | *protocol-number*}
12. **sequencing** {**transmit** | **receive** | **both**}
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class etherpw`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
Step 4	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.
Step 5	**protocol** {**l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw)# protocol l2tpv3 class1`	(Optional) Specifies the L2TPv3 signaling protocol to be used to manage the pseudowires created with the control channel parameters in the specified L2TP class (see the section "Configuring L2TP Control Channel Parameters"). • If the *l2tp-class-name* argument is not specified, the default values for L2TP control channel parameters are used. The default **protocol** option is **l2tpv3**. • If you do not want to use signaling in the L2TPv3 sessions created with this pseudowire class, enter **protocol none**.
Step 6	**ip local interface** *interface-name* **Example:** `Device(config-pw)# ip local interface e0/0`	Specifies the PE device interface whose IP address is to be used as the source IP address for sending tunneled packets. • The same or a different local interface name can be used for each of the pseudowire classes configured between a pair of PE devices. **Note** This command must be configured for pseudowire-class configurations using L2TPv3 as the data encapsulation method.
Step 7	**ip pmtu** **Example:** `Device(config-pw)# ip pmtu`	(Optional) Enables the discovery of the PMTU for tunneled traffic and helps fragmentation. • This command enables the processing of ICMP unreachable messages that indicate fragmentation errors in the backbone network that carries L2TPv3 session traffic. Also, this command enables MTU checking for IP packets sent into the session and that have the DF bit set. Any IP packet larger than the MTU is dropped and an ICMP unreachable message is sent. MTU discovery is disabled by default. **Note** The **ip pmtu** command is not supported if you disabled signaling with the **protocol none** command in Step 5. • This command must be enabled in the pseudowire class configuration to enable fragmentation of IP packets before the data enters the pseudowire.

	Command or Action	Purpose
		Note To enable fragmentation of IP packets before the data enters the pseudowire, Cisco recommends that you also enter the **ip dfbit set** command in pseudowire class configuration mode. This allows the PMTU to be obtained more rapidly. **Note** When the **ip pmtu** command is enabled, the DF bit is copied from the inner IP header to the outer IP header. If no IP header is found inside the Layer 2 frame, the DF bit in the outer IP is set to 0.
Step 8	**ip tos** {**value** *value* | **reflect**} **Example:** `Device(config-pw)# ip tos reflect`	(Optional) Configures the value of the ToS byte in IP headers of tunneled packets, or reflects the ToS byte value from the inner IP header. • Valid values for the *value* argument range from 0 to 255. The default ToS byte value is 0.
Step 9	**ip dfbit set** **Example:** `Device(config-pw)# ip dfbit set`	(Optional) Configures the value of the DF bit in the outer headers of tunneled packets. • Use this command if (for performance reasons) you do not want reassembly of tunneled packets on the peer PE device. • This command is disabled by default.
Step 10	**ip ttl** *value* **Example:** `Device(config-pw)# ip ttl 100`	(Optional) Configures the value of the time to live (TTL) byte in the IP headers of tunneled packets. • Valid values for the *value* argument range from 1 to 255. The default TTL byte value is 255.
Step 11	**ip protocol** {**l2tp** | *protocol-number*} **Example:** `Device(config-pw)# ip protocol l2tp`	(Optional) Configures the IP protocol to be used for tunneling packets.
Step 12	**sequencing** {**transmit** | **receive** | **both**} **Example:** `Device(config-pw)# sequencing both`	(Optional) Specifies the direction in which sequencing of data packets in a pseudowire is enabled: • **transmit—**Updates the Sequence Number field in the headers of data packets sent over the pseudowire according to the data encapsulation method that is used. • **receive—**Keeps the Sequence Number field in the headers of data packets received over the pseudowire. Out-of-order packets are dropped. • **both—**Enables both the **transmit** and **receive** options.
Step 13	**exit** **Example:** `Device(config-pw)# exit`	Exits pseudowire class configuration mode.

Configuring the Xconnect Attachment Circuit

The virtual circuit identifier that you configure creates the binding between a pseudowire configured on a PE device and an attachment circuit in a CE device. The virtual circuit identifier configured on the PE device at one end of the L2TPv3 control channel must also be configured on the peer PE device at the other end.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Device(config)# interface ethernet 0/0`	Specifies the interface by type (for example, Ethernet), slot, and port number, and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Device(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect`	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device. • At least one of the following pseudowire class parameters must be configured for the *pseudowire-parameters* argument: • **encapsulation** {**l2tpv3** [**manual**] | **mpls**}—Specifies the tunneling method used to encapsulate data in the pseudowire: ◦ **l2tpv3**—L2TPv3 is the tunneling method to be used.

	Command or Action	Purpose
		◦ **manual—**(Optional) No signaling is to be used in the L2TPv3 control channel. This command places the device in xconnect configuration mode for the manual configuration of L2TPv3 parameters for the attachment circuit. ◦ **mpls—**MPLS is the tunneling method to be used. • **pw-class** {*pw-class-name*}**—**The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. • The optional **encapsulation** parameter specifies the method of pseudowire tunneling used: L2TPv3 or MPLS. Enter **manual** if you do not want signaling to be used in the L2TPv3 control channel. The **encapsulation l2tpv3 manual** keyword combination enters xconnect configuration submode. See the section "*Manually Configuring L2TPv3 Session Parameters*" for the other L2TPv3 commands that you must enter to complete the configuration of the L2TPv3 control channel. If you do not enter an **encapsulation** value, the encapsulation method entered with the **password** command in the Configuring the Xconnect Attachment Circuit task is used. • The optional **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. Specify the pseudowire-class option if you need to configure more advanced options. **Note** You must configure either the **encapsulation** or the **pw-class** option or both. **Note** If you select L2TPv3 as your data encapsulation method, you must specify the **pw-class** keyword. • The optional **sequencing** parameter specifies whether sequencing is required for packets that are received, sent, or both received and sent.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring the Xconnect Attachment Circuit for ATM VP Mode Single Cell Relay over L2TPv3

The ATM VP Mode Single Cell Relay over L2TPv3 feature allows cells coming into a predefined PVP on the ATM interface to be transported over an L2TPv3 pseudowire to a predefined PVP on the egress ATM interface. This task binds a PVP to an L2TPv3 pseudowire for xconnect service.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **atm pvp** *vpi* [**l2transport**]
5. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*

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 slot / port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**atm pvp** *vpi* [**l2transport**] **Example:** `Router(config-if)# atm pvp 5 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells. • The **l2transport** keyword indicates that the PVP is for cell relay. After you enter this command, the router enters l2transport PVP configuration mode. This configuration mode is for Layer 2 transport only; it is not for terminated PVPs.
Step 5	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-if-atm-l2trans-pvp)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it.

Configuring the Xconnect Attachment Circuit for ATM Single Cell Relay VC Mode over L2TPv3

The ATM Single Cell Relay VC Mode over L2TPv3 feature maps one VCC to a single L2TPv3 session. All ATM cells arriving at an ATM interface with the specified VPI and VCI are encapsulated into a single L2TP packet.

The ATM Single Cell Relay VC mode feature can be used to carry any type of AAL traffic over the pseudowire. It will not distinguish OAM cells from User data cells. In this mode, PM and Security OAM cells are also transported over the pseudowire.

Perform this task to enable the ATM Single Cell Relay VC Mode over L2TPv3 feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **pvc** [*name*] *vpi* / *vci* **l2transport**
5. **encapsulation aal0**
6. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*

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 slot* / *port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 5/500 l2transport`	Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode. • The **l2transport** keyword indicates that the PVC is for Layer 2 switched connections. After you enter this command, the router enters ATM VC configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation aal0** **Example:** `Router(config-atm-vc)# encapsulation aal0`	Specifies ATM AAL0 encapsulation for the PVC.
Step 6	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-atm-vc)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.

Configuring the Xconnect Attachment Circuit for ATM Port Mode Cell Relay over L2TPv3

The ATM Port Mode Cell Relay feature packs ATM cells arriving at an ingress ATM interface into L2TPv3 data packets and transports them to the egress ATM interface. A single ATM cell is encapsulated into each L2TPv3 data packet.

Perform this task to enable the ATM Port Mode Cell Relay over L2TPv3 feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.

Configuring the Xconnect Attachment Circuit for ATM Cell Packing over L2TPv3

The ATM Cell Packing over L2TPv3 feature allows multiple ATM frames to be packed into a single L2TPv3 data packet. ATM cell packing can be configured for Port mode, VP mode, and VC mode. Perform one of the following tasks to configure the ATM Cell Packing over L2TPv3 feature:

Configuring Port Mode ATM Cell Packing over L2TPv3

Perform this task to configure port mode ATM cell packing over L2TPv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*]
5. **cell-packing** [*cells*] [**mcpt-timer** *timer*]
6. **xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]

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 slot / port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*] **Example:** `Router(config-if)# atm mcpt-timers 10 100 1000`	(Optional) Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an L2TPv3 packet.
Step 5	**cell-packing** [*cells*] [**mcpt-timer** *timer*] **Example:** `Router(config-if)# cell-packing 10 mcpt-timer 2`	Enables the packing of multiple ATM cells into each L2TPv3 data packet. • *cells* --(Optional) The number of cells to be packed into an L2TPv3 data packet. The default number of ATM cells to be packed is the maximum transmission unit (MTU) of the interface divided by 52. • **mcpt-timer** *timer* --(Optional) Specifies which maximum cell packing timeout (MCPT) timer to use. The MCPT timers are set using the **mcpt-timers** command. The default value is 1.

	Command or Action	Purpose
Step 6	**xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# xconnect 10.0.3.201 888 encapsulation l2tpv3`	Binds an attachment circuit to a Layer 2 pseudowire and enters xconnect configuration mode.

Configuring VP Mode ATM Cell Packing over L2TPv3

Perform this task to configure VP mode ATM cell packing over L2TPv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*]
5. **atm pvp** *vpi* [*peak-rate*] [**l2transport**]
6. **cell-packing** [*cells*] [**mcpt-timer** *timer*]
7. **xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]

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 slot / port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*] **Example:** `Router(config-if)# atm mcpt-timers 10 100 1000`	(Optional) Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an L2TPv3 packet.
Step 5	**atm pvp** *vpi* [*peak-rate*] [**l2transport**] **Example:** `Router(config-if)# atm pvp 10 l2transport`	Create a PVP used to multiplex (or bundle) one or more VCs.
Step 6	**cell-packing** [*cells*] [**mcpt-timer** *timer*] **Example:** `Router(config-if)# cell-packing 10 mcpt-timer 2`	Enables the packing of multiple ATM cells into each L2TPv3 data packet. • *cells* --(Optional) The number of cells to be packed into an L2TPv3 data packet. The default number of ATM cells to be packed is the MTU of the interface divided by 52. • **mcpt-timer** *timer* --(Optional) Specifies which MCPT timer to use. The MCPT timers are set using the **mcpt-timers** command. The default value is 1.
Step 7	**xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# xconnect 10.0.3.201 888 encapsulation l2tpv3`	Binds an attachment circuit to a Layer 2 pseudowire and enters xconnect configuration mode.

Configuring VC Mode ATM Cell Packing over L2TPv3

Perform this task to configure VC mode ATM cell packing over L2TPv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*]
5. **pvc** [*name*] *vpi* / *vci* [**ces** | **ilmi** | **qsaal** | **smds** | **l2transport**]
6. **encapsulation aal0**
7. **cell-packing** [*cells*] [**mcpt-timer** *timer*]
8. **xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]

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 slot / port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**atm mcpt-timers** [*timeout-value-1 timeout-value-2 timeout-value-3*] **Example:** `Router(config-if)# atm mcpt-timers 10 100 1000`	(Optional) Sets up the cell-packing timers, which specify how long the PE router can wait for cells to be packed into an L2TPv3 packet.
Step 5	**pvc** [*name*] *vpi* / *vci* [**ces**	**ilmi**	**qsaal**	**smds**	**l2transport**] **Example:** `Router(config-if)# pvc 1/32 l2transport`	Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode.

	Command or Action	Purpose
Step 6	**encapsulation aal0** **Example:** `Router(config-if-atm-vc)# encapsulation aal0`	Specifies ATM AAL0 encapsulation for the PVC.
Step 7	**cell-packing** [*cells*] [**mcpt-timer** *timer*] **Example:** `Router(config-if-atm-vc)# cell-packing 10 mcpt-timer 2`	Enables the packing of multiple ATM cells into each L2TPv3 data packet. • *cells* --(Optional) The number of cells to be packed into an L2TPv3 data packet. The default number of ATM cells to be packed is the MTU of the interface divided by 52. • **mcpt-timer** *timer* --(Optional) Specifies which timer to use. The mcpt timers are set using the **mcpt-timers** command. The default value is 1.
Step 8	**xconnect** *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if-atm-vc)# xconnect 10.0.3.201 888 encapsulation l2tpv3`	Binds an attachment circuit to a Layer 2 pseudowire and enters xconnect configuration mode.

Configuring the Xconnect Attachment Circuit for ATM AAL5 SDU Mode over L2TPv3

The ATM AAL5 SDU Mode feature maps the AAL5 payload of an AAL5 PVC to a single L2TPv3 session. This service will transport OAM and RM cells, but does not attempt to maintain the relative order of these cells with respect to the cells that comprise the AAL5 CPCS-PDU. OAM cells that arrive during the reassembly of a single AAL5 CPCS-PDU are sent immediately over the pseudowire, followed by the AAL5 SDU payload.

Beginning in Cisco IOS Release 12.0(30)S, you may choose to configure the ATM AAL5 SDU Mode feature in ATM VC configuration mode or in VC class configuration mode.

To enable the ATM AAL5 SDU Mode feature, perform one of the following tasks:

Configuring ATM AAL5 SDU Mode over L2TPv3 in ATM VC Configuration Mode

Perform this task to bind a PVC to an L2TPv3 pseudowire for ATM AAL5 SDU mode xconnect service.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **pvc** [*name*] *vpi* / *vci* [**l2transport**]
5. **encapsulation aal5**
6. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*

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 slot* / *port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi* / *vci* [**l2transport**] **Example:** `Router(config-if)# pvc 5/500 l2transport`	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode. • The **l2transport** keyword indicates that the PVC is for Layer 2 switched connections. After you enter this command, the router enters ATM VC configuration mode.
Step 5	**encapsulation aal5** **Example:** `Router(config-atm-vc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC.
Step 6	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel.

	Command or Action	Purpose
	Example: `Router(config-atm-vc)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	• The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**keyword binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.

Configuring ATM AAL5 SDU Mode over L2TPv3 in VC Class Configuration Mode

You can create a VC class that specifies AAL5 encapsulation and then attach the VC class to an interface, subinterface, or PVC. Perform this task to create a VC class configured for AAL5 encapsulation and attach the VC class to an interface.

Note This task requires Cisco IOS Release 12.0(30)S or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation aal5**
5. **end**
6. **interface** *type slot* / *port*
7. **class-int** *vc-class-name*
8. **pvc** [*name*] *vpi* / *vci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation l2tpv3**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm aal5class`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation aal5** **Example:** `Router(config-vc-class)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC.
Step 5	**end** **Example:** `Router(config-vc-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 6	**interface** *type slot / port* **Example:** `Router(config)# interface atm 1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 7	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int aal5class`	Applies a VC class on an the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	**pvc** [*name*] *vpi / vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode. • The **l2transport** keyword indicates that the PVC is for Layer 2 switched connections. After you enter this command, the router enters ATM VC configuration mode.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation l2tpv3**	Binds the attachment circuit to a pseudowire VC.

	Command or Action	Purpose
	Example: `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation l2tpv3`	

Configuring OAM Local Emulation for ATM AAL5 over L2TPv3

If a PE router does not support the transport of OAM cells across an L2TPv3 session, you can use OAM cell emulation to locally terminate or loopback the OAM cells. You configure OAM cell emulation on both PE routers. You use the **oam-ac emulation-enable** command 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 has been 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 have the following information cells:

- 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 as down and sends an RDI cell to let the remote end know about the failure.

Beginning in Cisco IOS Release 12.0(30)S, you may choose to configure the OAM Local Emulation for ATM AAL5 over L2TPv3 feature in ATM VC configuration mode or in VC class configuration mode.

To enable the OAM Local Emulation for ATM AAL5 over L2TPv3 feature, perform one of the following tasks:

Configuring OAM Local Emulation for ATM AAL5 over L2TPv3 in ATM VC Configuration Mode

Perform this task to enable the OAM Local Emulation for ATM AAL5 over L2TPv3 feature in ATM VC configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **pvc** [*name*] *vpi* / *vci* [**l2transport**]
5. **encapsulation aal5**
6. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*
7. **oam-ac emulation-enable** [*ais-rate*]
8. **oam-pvc manage** [*frequency*]

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 slot* / *port* **Example:** `Router(config)# interface ATM 4/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi* / *vci* [**l2transport**] **Example:** `Router(config-if)# pvc 5/500 l2transport`	Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode. • The **l2transport** keyword indicates that the PVC is for Layer 2 switched connections. After you enter this command, the router enters ATM VC configuration mode.
Step 5	**encapsulation aal5** **Example:** `Router(config-atm-vc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC.
Step 6	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel.

	Command or Action	Purpose
	Example: `Router(config-atm-vc)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	• The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.
Step 7	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-atm-vc)# oam-ac emulation-enable 30`	Enables OAM cell emulation on AAL5 over L2TPv3. • The **oam-ac emulation-enable** command lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.
Step 8	**oam-pvc manage** [*frequency*] **Example:** `Router(config-atm-vc)# oam-pvc manage`	(Optional) Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. • The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds. **Note** You can configure the **oam-pvc manage** command only after you issue the **oam-ac emulation-enable** command.

Configuring OAM Local Emulation for ATM AAL5 over L2TPv3 in VC Class Configuration Mode

This task configures OAM Cell Emulation as part of a VC class. After a VC class is configured, you can apply the VC class to an interface, a subinterface, or a VC.

When you apply a VC class to an interface, the settings in the VC class apply to all the VCs on that interface unless you specify otherwise at a lower level, such as the subinterface or VC level. For example, if you create a VC class that specifies OAM cell emulation and sets the AIS cell rate to 30 seconds and apply that VC class to an interface, every VC on that interface will use the AIS cell rate of 30 seconds. If you then enable OAM cell emulation on a single PVC and set the AIS cell rate to 15 seconds, the 15 second AIS cell rate configured at the PVC level will take precedence over the 30 second AIS cell rate configured at the interface level.

Perform this task to create a VC class configured for OAM emulation and to attach the VC class to an interface.

Note This task requires Cisco IOS Release 12.0(30)S or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** layer-type
5. **oam-ac emulation-enable** [*ais-rate*]
6. **oam-pvc manage** [*frequency*]
7. **end**
8. **interface** *type slot* / *port*
9. **class-int** *vc-class-name*
10. **pvc** [*name*] *vpi* / *vci* **l2transport**
11. **xconnect** *peer-router-id vcid* **encapsulation l2tpv3**

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	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm oamclass`	Creates a VC class and enters vc-class configuration mode.
Step 4	**encapsulation** layer-type **Example:** `Router(config-vc-class)# encapsulation aal5`	Configures the ATM adaptation layer (AAL) and encapsulation type.
Step 5	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-vc-class)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over L2TPv3. • The *ais-rate* argument lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.

	Command or Action	Purpose
Step 6	**oam-pvc manage** [*frequency*] **Example:** `Router(config-vc-class)# oam-pvc manage`	(Optional) Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. • The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds. **Note** You can configure the **oam-pvc manage** command only after you issue the **oam-ac emulation-enable** command.
Step 7	**end** **Example:** `Router(config-vc-class)# end`	Ends your configuration session by exiting to privileged EXEC mode.
Step 8	**interface** *type slot* / *port* **Example:** `Router(config)# interface atm1/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 9	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int oamclass`	Applies a VC class on an the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 10	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM VC configuration mode. • The **l2transport** keyword indicates that the PVC is for Layer 2 switched connections. After you enter this command, the router enters ATM VC configuration mode.
Step 11	**xconnect** *peer-router-id vcid* **encapsulation l2tpv3** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation l2tpv3`	Binds the attachment circuit to a pseudowire VC.

Configuring Protocol Demultiplexing for L2TPv3

Configuring Protocol Demultiplexing for Ethernet Interfaces

Perform this task to configure the Protocol Demultiplexing feature on an Ethernet interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **ip address** *ip-address mask* [**secondary**]
5. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*
6. **match protocol ipv6**
7. **exit**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Device(config)# interface ethernet 0/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.128.4`	Sets a primary or secondary IP address for an interface.
Step 5	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Device(config-if)# xconnect 10.0.3.201 888 pw-class demux`	Specifies the IP address of the peer PE device and the 32-bit VCI shared between the PE at each end of the control channel, and enters xconnect configuration mode. • The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device.

	Command or Action	Purpose
		• **pw-class** *pw-class-name*—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the section "*Manually Configuring L2TPv3 Session Parameters*" for information about manually configuring the L2TPv3 session parameters.
Step 6	**match protocol ipv6** **Example:** `Device(config-if-xconn)# match protocol ipv6`	Enables protocol demultiplexing of IPv6 traffic.
Step 7	**exit** **Example:** `Device(config-if-xconn)# exit`	Exits xconnect configuration mode.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring Protocol Demultiplexing for Frame Relay Interfaces

Perform this task to configure the Protocol Demultiplexing feature on a Frame Relay interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port-adapter* **.** *subinterface-number* [**multipoint** | **point-to-point**]
4. **ip address** *ip-address mask* [**secondary**]
5. **frame-relay interface-dlci** *dlci* [**ietf** | **cisco**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*]
6. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*
7. **match protocol ipv6**

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 slot / port-adapter . subinterface-number* [**multipoint** | **point-to-point**] **Example:** `Router(config)# interface serial 1/1.2 multipoint`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 172.16.128.4`	Sets a primary or secondary IP address for an interface.
Step 5	**frame-relay interface-dlci** *dlci* [**ietf** | **cisco**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*] **Example:** `Router(config-if)# frame-relay interface-dlci 100`	Assigns a DLCI to a specified Frame Relay subinterface on the router or access server, assigns a specific PVC to a DLCI, or applies a virtual template configuration for a PPP session and enters Frame Relay DLCI interface configuration mode.
Step 6	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-fr-dlci)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel and enters xconnect configuration mode. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it.

	Command or Action	Purpose
		Note The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.
Step 7	**match protocol ipv6** **Example:** `Router(config-if-xconn)# match protocol ipv6`	Enables protocol demultiplexing of IPv6 traffic.

Configuring Protocol Demultiplexing for PPP Interfaces

Perform this task to configure the Protocol Demultiplexing feature on a Point-to-Point Protocol (PPP) interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **ip address** *ip-address mask* [**secondary**]
5. **encapsulation** *physical-interface*
6. **ppp** *interface-address*
7. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*
8. **match protocol ipv6**

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	**interface** *type slot / port* **Example:** `Router(config)# interface serial 0/1`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.167.1.1 255.255.255.252`	Sets a primary or secondary IP address for an interface.
Step 5	**encapsulation** *physical-interface* **Example:** `Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation for IPv6.
Step 6	**ppp** *interface-address* **Example:** `Router(config-if)# ppp ipv6cp id proxy A8BB:CCFF:FE00:7000`	
Step 7	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel and enters xconnect configuration mode. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class**parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section for information about manually configuring the L2TPv3 session parameters.
Step 8	**match protocol ipv6** **Example:** `Router(config-if-xconn)# match protocol ipv6`	Enables protocol demultiplexing of IPv6 traffic.

Configuring Protocol Demultiplexing for HDLC Interfaces

Perform this task to configure the Protocol Demultiplexing feature on a High-Level Data Link Control (HDLC) interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **ip address** *ip-address mask* [**secondary**]
5. **xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name*
6. **match protocol ipv6**

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 slot / port* **Example:** `Router(config)# interface serial 0/0`	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 172.16.128.4 255.255.255.252`	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 5	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.3.201 888 pw-class atm-xconnect`	Specifies the IP address of the peer PE router and the 32-bit VCI shared between the PE at each end of the control channel and enters xconnect configuration mode. • The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The **pw-class** parameter binds the xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. **Note** The L2TPv3 session can also be provisioned manually. See the *Manually Configuring L2TPv3 Session Parameters* section or information about manually configuring the L2TPv3 session parameters.
Step 6	**match protocol ipv6** **Example:** `Router(config-if-xconn)# match protocol ipv6`	Enables protocol demultiplexing of IPv6 traffic.

Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The L2TPv3 Custom Ethertype for dot1q and QinQ Encapsulations feature lets you configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with QinQ or dot1Q encapsulations. You can set the custom Ethertype to 0x9100, 0x9200, or 0x88A8. To define the Ethertype field type, you use the **dot1q tunneling ethertype** command.

Perform this task to set a custom Ethertype.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dot1q tunneling ethertype** {**0x88A8** | **0x9100** | **0x9200**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**dot1q tunneling ethertype** {**0x88A8** | **0x9100** | **0x9200**} **Example:** `Device(config-if)# dot1q tunneling ethertype 0x9100`	Defines the Ethertype field type used by peer devices when implementing Q-in-Q VLAN tagging.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Manually Clearing L2TPv3 Tunnels

Perform this task to manually clear a specific L2TPv3 tunnel and all the sessions in that tunnel.

SUMMARY STEPS

1. **enable**
2. **clear l2tun** {**l2tp-class** *l2tp-class-name* \| **tunnel id** *tunnel-id* \| **local ip** *ip-address* \| **remote ip** *ip-address* \| **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**clear l2tun** {**l2tp-class** *l2tp-class-name* | **tunnel id** *tunnel-id* | **local ip** *ip-address* | **remote ip** *ip-address* | **all**} **Example:** `Device# clear l2tun tunnel id 56789`	Clears the specified L2TPv3 tunnel. (This command is not available if there are no L2TPv3 tunnel sessions configured.) • **l2tp-class** *l2tp-class-name*—All L2TPv3 tunnels with the specified L2TP class name are torn down. • **tunnel id** *tunnel-id*—The L2TPv3 tunnel with the specified tunnel ID are torn down. • **local ip** *ip-address*—All L2TPv3 tunnels with the specified local IP address are torn down. • **remote ip** *ip-address*—All L2TPv3 tunnels with the specified remote IP address are torn down. • **all**—All L2TPv3 tunnels are torn down.

Configuration Examples for Layer 2 Tunneling Protocol Version 3

Note The IP addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Example: Configuring a Static L2TPv3 Session for an Xconnect Ethernet Interface

L2TPv3 is the only encapsulation method that supports a manually provisioned session setup. This example shows how to configure a static session configuration in which all control channel parameters are set up in advance. There is no control plane used and no negotiation phase to set up the control channel. The PE device starts sending tunneled traffic as soon as the Ethernet interface (int e0/0) comes up. The virtual circuit identifier, 123, is not used. The PE sends L2TP data packets with session ID 111 and cookie 12345. In turn, the PE expects to receive L2TP data packets with session ID 222 and cookie 54321.

```
l2tp-class l2tp-defaults
 retransmit initial retries 30
 cookie-size 8
pseudowire-class ether-pw
```

```
 encapsulation l2tpv3
 protocol none
 ip local interface Loopback0
interface Ethernet 0/0
 xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
 l2tp id 222 111
 l2tp cookie local 4 54321
 l2tp cookie remote 4 12345
 l2tp hello l2tp-defaults
```

Example: Configuring a Negotiated L2TPv3 Session for an Xconnect VLAN Subinterface

The following is a sample configuration of a dynamic L2TPv3 session for a VLAN xconnect interface. In this example, only VLAN traffic with a VLAN ID of 5 is tunneled. In the other direction, the L2TPv3 session identified by a virtual circuit identifier of 123 receives forwarded frames whose VLAN ID fields are rewritten to contain the value 5. L2TPv3 is used as both the control plane protocol and the data encapsulation.

```
l2tp-class class1
 authentication
 password secret
pseudowire-class vlan-xconnect
 encapsulation l2tpv3
 protocol l2tpv3 class1
 ip local interface Loopback0
interface Ethernet0/0.1
 encapsulation dot1q 5
 xconnect 10.0.3.201 123 pw-class vlan-xconnect
```

Example: Configuring a Negotiated L2TPv3 Session for Local HDLC Switching

The following is a sample configuration of a dynamic L2TPv3 session for local HDLC switching. In this example, note that it is necessary to configure two different IP addresses at the endpoints of the L2TPv3 pseudowire because the virtual circuit identifier must be unique for a given IP address.

```
interface loopback 1
 ip address 10.0.0.1 255.255.255.255
interface loopback 2
 ip address 10.0.0.2 255.255.255.255
pseudowire-class loopback1
 encapsulation l2tpv3
 ip local interface loopback1
pseudowire-class loopback2
 encapsulation l2tpv3
 ip local interface loopback2
interface s0/0
 encapsulation hdlc
 xconnect 10.0.0.1 100 pw-class loopback2
interface s0/1
 encapsulation hdlc
 xconnect 10.0.0.2 100 pw-class loopback1
```

Example: Verifying an L2TPv3 Session

- To display information about current L2TPv3 sessions on a router, use the **show l2tun session brief** command:

```
Router# show l2tun session brief
L2TP Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    State     Username, Intf/
                                      sess/cir  Vcid, Circuit
2391726297 2382731778 6.6.6.6         est,UP    100, Gi0/2/0
```

- To display detailed information about current L2TPv3 sessions on a router, use the **show l2tun session all** command:

```
Router# show l2tun session all
Session Information Total tunnels 0 sessions 1
Session id 111 is up, tunnel id 0
Call serial number is 0
Remote tunnel name is
  Internet address is 10.0.0.1
  Session is manually signalled
  Session state is established, time since change 00:06:05
    0 Packets sent, 0 received
    0 Bytes sent, 0 received
    Receive packets dropped:
      out-of-order:             0
      total:                    0
    Send packets dropped:
      exceeded session MTU:     0
      total:                    0
  Session vcid is 123
  Session Layer 2 circuit, type is ATM VPC CELL, name is ATM3/0/0:1000007
  Circuit state is UP
    Remote session id is 222, remote tunnel id 0
  DF bit off, ToS reflect disabled, ToS value 0, TTL value 255
  Session cookie information:
    local cookie, size 8 bytes, value 00 00 00 00 00 00 00 64
    remote cookie, size 8 bytes, value 00 00 00 00 00 00 00 C8
  SSS switching enabled
  Sequencing is off
```

Example: Verifying an L2TP Control Channel

The L2TP control channel is used to negotiate capabilities, monitor the health of the peer PE router, and set up various components of an L2TPv3 session. To display information the L2TP control channels that are set up to other L2TP-enabled devices for all L2TP sessions on the router, use the **show l2tun tunnel** command.

```
Router# show l2tun tunnel
L2TP Tunnel Information Total tunnels 1 sessions 1
LocTunID   RemTunID   Remote Name   State  Remote Address  Sessn L2TP Class/
                                                          Count VPDN Group
2382731778 2280318174 l2tp-asr-2    est    6.6.6.6         1     l2tp_default_cl
```

To display detailed information the L2TP control channels that are set up to other L2TP-enabled devices for all L2TP sessions on the router, use the **show l2tun tunnel all** command.

```
Router# show l2tun tunnel all
Tunnel id 26515 is up, remote id is 41814, 1 active sessions
  Tunnel state is established, time since change 03:11:50
  Tunnel transport is IP (115)
  Remote tunnel name is tun1
    Internet Address 172.18.184.142, port 0
  Local tunnel name is Router
    Internet Address 172.18.184.116, port 0
```

```
 Tunnel domain is
 VPDN group for tunnel is
 0 packets sent, 0 received
 0 bytes sent, 0 received
 Control Ns 11507, Nr 11506
 Local RWS 2048 (default), Remote RWS 800
 Tunnel PMTU checking disabled
 Retransmission time 1, max 1 secondsPF
 Unsent queuesize 0, max 0
 Resend queuesize 1, max 1
 Total resends 0, ZLB ACKs sent 11505
 Current nosession queue check 0 of 5
 Retransmit time distribution: 0 0 0 0 0 0 0 0 0 0
 Sessions disconnected due to lack of resources 0
```

Example: Configuring L2TPv3 Control Channel Authentication

The following example shows how to configure CHAP-style authentication of the L2TPv3 control channel:

```
l2tp-class class0
 authentication
 password cisco
```

The following example shows how to configure control channel authentication using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class1
 digest secret cisco hash sha
 hidden
```

The following example shows how to configure control channel integrity checking and how to disable validation of the message digest using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class2
 digest hash sha
 no digest check
```

The following example shows how to disable the validation of the message digest using the L2TPv3 Control Message Hashing feature:

```
l2tp-class class3
 no digest check
```

Example: Configuring L2TPv3 Digest Secret Graceful Switchover

The following example shows how to use the L2TPv3 Digest Secret Graceful Switchover feature to change the L2TP control channel authentication password for the L2TP class named class1. This example assumes that you already have an old password configured for the L2TP class named class1.

```
Device(config)# l2tp-class class1
Device(config-l2tp-class)# digest secret cisco2 hash sha
!
! Verify that all peer PE devices have been updated to use the new password before
! removing the old password.
!
Device(config-l2tp-class)# no digest secret cisco hash sha
```

Example: Verifying L2TPv3 Digest Secret Graceful Switchover

The following **show l2tun tunnel all** command output shows information about the L2TPv3 Digest Secret Graceful Switchover feature:

```
Device# show l2tun tunnel all
! The output below displays control channel password information for a tunnel which has
! been updated with the new control channel authentication password.
!
Tunnel id 12345 is up, remote id is 54321, 1 active sessions
```

```
Control message authentication is on, 2 secrets configured
Last message authenticated with first digest secret
!
! The output below displays control channel password information for a tunnel which has
! only a single control channel authentication password configured.
!
Tunnel id 23456 is up, remote id is 65432, 1 active sessions
!
Control message authentication is on, 1 secrets configured
Last message authenticated with first digest secret
!
! The output below displays control channel password information for a tunnel which is
! communicating with a peer that has only the new control channel authentication password
! configured.
!
Tunnel id 56789 is up, remote id is 98765, 1 active sessions
!
Control message authentication is on, 2 secrets configured
Last message authenticated with second digest secret
```

Example: Configuring a Pseudowire Class for Fragmentation of IP Packets

The following is a sample configuration of a pseudowire class that will allow IP traffic generated from the CE device to be fragmented before entering the pseudowire:

```
pseudowire class class1
 encapsulation l2tpv3
 ip local interface Loopback0
 ip pmtu
 ip dfbit set
```

Configuring ATM VP Mode Single Cell Relay over L2TPv3 Example

The following configuration binds a PVP to an xconnect attachment circuit to forward ATM cells over an established L2TPv3 pseudowire:

```
pw-class atm-xconnect
 encapsulation l2tpv3
interface ATM 4/1
 atm pvp 5 l2transport
 xconnect 10.0.3.201 888 pw-class atm-xconnect
```

Verifying ATM VP Mode Single Cell Relay over L2TPv3 Configuration Example

To verify the configuration of a PVP, use the **show atm vp** command in privileged EXEC mode:

```
Router#
show atm vp 5
ATM4/1/0  VPI: 5, Cell-Relay, PeakRate: 155000, CesRate: 0, DataVCs: 0,
CesVCs: 0, Status: ACTIVE
  VCD    VCI Type     InPkts  OutPkts  AAL/Encap     Status
    8      3 PVC           0        0  F4 OAM        ACTIVE
    9      4 PVC           0        0  F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0,
TotalBroadcasts: 0
```

Configuring ATM Single Cell Relay VC Mode over L2TPv3 Example

The following example shows how to configure the ATM Single Cell Relay VC Mode over L2TPv3 feature:

```
pw-class atm-xconnect
 encapsulation l2tpv3
interface ATM 4/1
 pvc 5/500 l2transport
  encapsulation aal0
  xconnect 10.0.3.201 888 pw-class atm-xconnect
```

Verifying ATM Single Cell Relay VC Mode over L2TPv3 Example

The following **show atm vc**command output displays information about VCC cell relay configuration:

```
Router#
show atm vc
VCD/                                                 Peak   Avg/Min    Burst
Interface   Name  VPI   VCI   Type     Encaps       Kbps    Kbps      Cells          Sts
2/0         4      9    901    PVC      AAL0     149760    N/A                       UP
```

The following **show l2tun session** command output displays information about VCC cell relay configuration:

```
Router#
show l2tun session all
Session Information Total tunnels 1 sessions 2
Session id 41883 is up, tunnel id 18252
Call serial number is 3211600003
Remote tunnel name is khur-l2tp
  Internet address is 10.0.0.2
  Session is L2TP signalled
  Session state is established, time since change 00:00:38
    8 Packets sent, 8 received
    416 Bytes sent, 416 received
    Receive packets dropped:
      out-of-order:             0
      total:                    0
    Send packets dropped:
      exceeded session MTU:     0
      total:                    0
  Session vcid is 124
  Session Layer 2 circuit, type is ATM VCC CELL, name is ATM2/0:9/901
  Circuit state is UP
    Remote session id is 38005, remote tunnel id 52436
  DF bit off, ToS reflect disabled, ToS value 0, TTL value 255
  No session cookie information available
  FS cached header information:
    encap size = 24 bytes
    00000000 00000000 00000000 00000000
    00000000 00000000
  Sequencing is off
```

Configuring ATM Port Mode Cell Relay over L2TPv3 Example

The following example shows how to configure the ATM Port Mode Cell Relay over L2TPv3 feature:

```
pw-class atm-xconnect
 encapsulation l2tpv3
interface atm 4/1
 xconnect 10.0.3.201 888 pw-class atm-xconnect
```

Configuring ATM Cell Packing over L2TPv3 Examples

The following examples show how to configure the ATM Cell Packing over L2TPv3 feature for Port mode, VP mode, and VC mode:

Port Mode

```
interface atm 4/1
 atm mcpt-timers 10 100 1000
 cell-packing 10 mcpt-timer 2
 xconnect 10.0.3.201 888 encapsulation l2tpv3
```

VP Mode

```
interface atm 4/1
 atm mcpt-timers 10 100 1000
 atm pvp 10 l2transport
 cell-packing 10 mcpt-timer 2
 xconnect 10.0.3.201 888 encapsulation l2tpv3
```

VC Mode

```
interface atm 4/1
 atm mcpt-timers 10 100 1000
 pvc 1/32 l2transport
  encapsulation aal0
  cell-packing 10 mcpt-timer 2
  xconnect 10.0.3.201 888 encapsulation l2tpv3
```

Configuring ATM AAL5 SDU Mode over L2TPv3 Examples

Configuring ATM AAL5 SDU Mode over L2TPv3 in ATM VC Configuration Mode

The following configuration binds a PVC to an xconnect attachment circuit to forward ATM cells over an established L2TPv3 pseudowire:

```
pw-class atm-xconnect
 encapsulation l2tpv3
interface atm 4/1
 pvc 5/500 l2transport
  encapsulation aal5
  xconnect 10.0.3.201 888 pw-class atm-xconnect
```

Configuring ATM AAL5 SDU Mode over L2TPv3 in VC-Class Configuration Mode

The following example configures ATM AAL5 over L2TPv3 in VC class configuration mode. The VC class is then applied to an interface.

```
vc-class atm aal5class
 encapsulation aal5
!
interface atm 1/0
 class-int aal5class
 pvc 1/200 l2transport
  xconnect 10.13.13.13 100 encapsulation l2tpv3
```

Verifying ATM AAL5 SDU Mode over L2TPv3 Configuration Examples

Verifying ATM AAL5 over MPLS in ATM VC Configuration Mode

To verify the configuration of a PVC, use the **show atm vc** command in privileged EXEC mode:

```
Router#
show atm vc
VCD/                                                 Peak   Avg/Min    Burst
Interface   Name  VPI   VCI   Type     Encaps       Kbps    Kbps      Cells         Sts
2/0         pvc    9    900    PVC      AAL5        2400    200                      UP
2/0         4      9    901    PVC      AAL5      149760    N/A                      UP
```

The following **show l2tun session** command output displays information about ATM VC mode configurations:

```
Router#
show l2tun session brief
Session Information Total tunnels 1 sessions 2
LocID      TunID      Peer-address    State        Username, Intf/
                                     sess/cir        Vcid, Circuit
41875      18252      10.0.0.2         est,UP       124, AT2/0:9/901
111           0       10.0.0.2         est,UP       123, AT2/0:9/900
```

Verifying ATM AAL5 over MPLS in VC Class Configuration Mode

To verify that ATM AAL5 over L2TPv3 is configured as part of a VC class, issue the **show atm class-links**command. The command output shows the type of encapsulation and that the VC class was applied to an interface.

```
Router#
show atm class links 1/100
Displaying vc-class inheritance for ATM1/0.0, vc 1/100:
no broadcast - Not configured - using default
encapsulation aal5 - VC-class configured on main interface
.
.
.
```

Configuring OAM Local Emulation for ATM AAL5 over L2TPv3 Examples

Configuring OAM Cell Emulation for ATM AAL5 over L2TPv3 in ATM VC Configuration Mode

The following configuration binds a PVC to an xconnect attachment circuit to forward ATM AAL5 frames over an established L2TPv3 pseudowire, enables OAM local emulation, and specifies that AIS cells are sent every 30 seconds:

```
pw-class atm-xconnect
 encapsulation l2tpv3
interface ATM 4/1
 pvc 5/500 l2transport
  encapsulation aal5
  xconnect 10.0.3.201 888 pw-class atm-xconnect
  oam-ac emulation-enable 30
```

Configuring OAM Cell Emulation for ATM AAL5 over L2TPv3 in VC Class Configuration Mode

The following example configures OAM cell emulation for ATM AAL5 over L2TPv3 in VC class configuration mode. The VC class is then 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 10.13.13.13 100 encapsulation l2tpv3
```

The following example configures OAM cell emulation for ATM AAL5 over L2TPv3 in VC class configuration mode. The VC class is then 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 10.13.13.13 100 encapsulation l2tpv3
```

The following example configures OAM cell emulation for ATM AAL5 over L2TPv3 in VC class configuration mode. The OAM cell emulation AIS rate is set to 30 for the VC class. 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.

```
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 10.13.13.13 100 encapsulation l2tpv3
```

Verifying OAM Local Emulation for ATM AAL5 over L2TPv3 Configuration Examples

The following **show atm pvc** command output shows that OAM cell emulation is enabled and working on the ATM PVC:

```
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 Protocol Demultiplexing for L2TPv3 Examples

The following examples show how to configure the Protocol Demultiplexing feature on the IPv4 PE routers. The PE routers facing the IPv6 network do not require IPv6 configuration.

Ethernet Interface

```
interface ethernet 0/1
 ip address 172.16.128.4
 xconnect 10.0.3.201 888 pw-class demux
  match protocol ipv6
```

Frame Relay Interface

```
interface serial 1/1.1 multipoint
 ip address 172.16.128.4
 frame-relay interface-dlci 100
  xconnect 10.0.3.201 888 pw-class atm-xconnect
   match protocol ipv6
```

PPP Interface

```
interface serial 0/0
 ip address 192.167.1.1 2555.2555.2555.252
 encapsulation ppp
 ppp ipv6cp id proxy A8BB:CCFF:FE00:7000
 xconnect 75.0.0.1 1 pw-class 12tp
  match protocol ipv6
```

HDLC Interface

```
interface serial 0/0
 ip address 192.168.1.2 2555.2555.2555.252
 xconnect 75.0.0.1 1 pw-class 12tp
  match protocol ipv6
```

Example: Manually Clearing an L2TPv3 Tunnel

The following example demonstrates how to manually clear a specific L2TPv3 tunnel using the tunnel ID:

```
clear l2tun tunnel 65432
```

Configuring Frame Relay DLCI-to-DLCI Switching Example

The following is a sample configuration for switching a Frame Relay DLCI over a pseudowire:

```
pseudowire-class fr-xconnect
 encapsulation l2tpv3
 protocol l2tpv3
 ip local interface Loopback0
 sequencing both
!
interface Serial0/0
 encapsulation frame-relay
 frame-relay intf-type dce
!
```

```
connect one Serial0/0 100 l2transport
 xconnect 10.0.3.201 555 pw-class fr-xconnect
!
connect two Serial0/0 200 l2transport
 xconnect 10.0.3.201 666 pw-class fr-xconnect
```

Configuring Frame Relay Trunking Example

The following is a sample configuration for setting up a trunk connection for an entire serial interface over a pseudowire. All incoming packets are switched to the pseudowire regardless of content.

Note that when you configure trunking for a serial interface, the trunk connection does not require an encapsulation method. You do not, therefore, need to enter the **encapsulation frame-relay** command. Reconfiguring the default encapsulation removes all xconnect configuration settings from the interface.

```
interface Serial0/0
 xconnect 10.0.3.201 555 pw-class serial-xconnect
```

Configuring QoS for L2TPv3 on the Cisco 7500 Series Example

The following example shows the MQC commands used on a Cisco 7500 series router to configure a CIR guarantee of 256 kbps on DLCI 100 and 512 kbps for DLCI 200 on the egress side of a Frame Relay interface that is also configured for L2TPv3 tunneling:

```
ip cef distributed
 class-map dlci100
 match fr-dlci 100
 class-map dlci200
 match fr-dlci 200
!
policy-map dlci
 class dlci100
 bandwidth 256
 class dlci200
 bandwidth 512
!
interface Serial0/0
 encapsulation frame-relay
 frame-relay interface-type dce
 service-policy output dlci
!
connect one Serial0/0 100 l2transport
 xconnect 10.0.3.201 555 encapsulation l2tpv3 pw-class mqc
!
connect two Serial0/0 200 l2transport
 xconnect 10.0.3.201 666 encapsulation l2tpv3 pw-class mqc
```

Configuring QoS for L2TPv3 on the Cisco 12000 Series Examples

Configuring QoS on a Frame Relay Interface in a TSC-Based L2TPv3 Tunnel Session

To apply a QoS policy for L2TPv3 to a Frame Relay interface on a Cisco 12000 series 2-port Channelized OC-3/STM-1 (DS1/E1) or 6-port Channelized T3 line card in a tunnel server card-based L2TPv3 tunnel session, you must:

- Use the **map-class frame-relay** *class-name* command in global configuration mode to apply a QoS policy to a Frame Relay class of traffic.
- Use the **frame-relay interface-dcli** *dcli-number* **switched** command (in interface configuration mode) to enter Frame Relay DLCI interface configuration mode and then the **class** command to configure a QoS policy for a Frame Relay class of traffic on the specified DLCI. You must enter a separate series of these configuration commands to configure QoS for each Frame Relay DLCI on the interface.

As shown in the following example, when you configure QoS for L2TPv3 on the ingress side of a Cisco 12000 series Frame Relay interface, you may also configure the value of the ToS byte used in IP headers of tunneled packets when you configure the L2TPv3 pseudowire (see the section Configuring the L2TPv3 Pseudowire).

The following example shows the MQC commands and ToS byte configuration used on a Cisco 12000 series router to apply a QoS policy for DLCI 100 on the ingress side of a Frame Relay interface configured for server card-based L2TPv3 tunneling:

```
policy-map frtp-policy
 class class-default
 police cir 8000 bc 6000 pir 32000 be 4000 conform-action transmit exceed-action
set-frde-transmit violate-action drop
!
map-class frame-relay fr-map
 service-policy input frtp-policy
!
interface Serial0/1/1:0
 encapsulation frame-relay
 frame-relay interface-dlci 100 switched
   class fr-map
 connect frol2tp1 Serial0/1/1:0 100 l2transport
   xconnect 10.0.3.201 666 encapsulation l2tpv3 pw-class aaa
!
pseudowire-class aaa
encapsulation l2tpv3
 ip tos value 96
```

To apply a QoS policy for L2TPv3 to the egress side of a Frame Relay interface on a Cisco 12000 series 2-port Channelized OC-3/STM-1 (DS1/E1) or 6-port Channelized T3 line card, you must:

- Use the **match ip precedence** command in class-map configuration mode to configure the IP precedence value used to determine the egress queue for each L2TPv3 packet with a Frame Relay payload.
- Use the **random-detect**command in policy-map class configuration mode to enable a WRED drop policy for a Frame Relay traffic class that has a bandwidth guarantee. Use the **random-detect precedence** command to configure the WRED and MDRR parameters for particular IP precedence values.

The next example shows the MQC commands used on a Cisco 12000 series Internet router to apply a QoS policy with WRED/MDRR settings for specified IP precedence values to DLCI 100 on the egress side of a Frame Relay interface configured for a server card-based L2TPv3 tunnel session:

```
class-map match-all d2
 match ip precedence 2
class-map match-all d3
 match ip precedence 3
!
policy-map o
 class d2
   bandwidth percent 10
   random-detect
   random-detect precedence 1 200 packets 500 packets 1
 class d3
   bandwidth percent 10
   random-detect
   random-detect precedence 1 1 packets 2 packets 1
!
```

```
map-class frame-relay fr-map
 service-policy output o
!
interface Serial0/1/1:0
 encapsulation frame-relay
 frame-relay interface-dlci 100 switched
 class fr-map
connect frol2tp1 Serial0/1/1:0 100 l2transport
 xconnect 10.0.3.201 666 encapsulation l2tpv3 pw-class aaa
```

Configuring Traffic Policing on an ISE E5 Interface in a Native L2TPv3 Tunnel Session

Starting in Cisco IOS Release 12.0(30)S, QoS traffic policing is supported on the following types of Edge Engine (ISE/E5) ingress interfaces bound to a native L2TPv3 tunnel session:

- ATM
- Frame Relay DLCIs

QoS traffic shaping in a native L2TPv3 tunnel session is supported on ATM ISE/E5 egress interfaces for the following service categories:

- UBR (unspecified bit rate)
- VBR-nrt (variable bit rate nonreal-time)

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or classes of service (CoS). The dual rate, 3-Color Marker in color-aware and color-blind modes, as defined in RFC 2698 for traffic policing, is supported on ingress ISE/E5 interfaces to classify packets.

The **police** command configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR). The following conform, exceed, and violate values for the *actions*argument are supported with the **police** command in policy-map configuration mode on an ISE/E5 interface bound to an L2TPv3 tunnel session:

- **conform-action** *actions* : Actions taken on packets that conform to the CIR and PIR.
  - **set-prec-tunnel**:Sets the IP precedence value in the tunnel header of a packet encapsulated for native L2TPv3 tunneling.
  - **set-dscp-tunnel**:Sets the IP differentiated services code point (DSCP) value in the tunnel header of a packet encapsulated for native L2TPv3 tunneling.
  - **transmit**:Sends the packet with no alteration.
- **exceed-action** *actions* : Actions taken on packets that conform to the CIR but not the PIR.
  - **drop**:Drops the packet.
  - **set-clp**(ATM only):Sets the Cell Loss Priority (CLP) bit from 0 to 1 in an ATM cell encapsulated for native L2TPv3 tunneling.
  - **set-dscp-tunnel**:Sets the DSCP value in the tunnel header of a packet encapsulated for native L2TPv3 tunneling.
  - **set-dscp-tunnel** and **set-clp**(ATM only): Sets the DSCP value in the tunnel header and the CLP bit in an ATM cell encapsulated for native L2TPv3 tunneling.

- **set-dscp-tunnel** and **set-frde**(Frame Relay only):Sets the DSCP value in the tunnel header and discard eligible (DE) bit in a Frame Relay packet encapsulated for native L2TPv3 tunneling.
- **set-frde**(Frame Relay only):Sets the DE bit in a Frame Relay packet encapsulated for native L2TPv3 tunneling.
- **set-prec-tunnel** and **set-clp**(ATM only):Sets the precedence value in the tunnel header and the CLP bit in an ATM cell encapsulated for native L2TPv3 tunneling.
- **set-prec-tunnel** and **set-frde**(Frame Relay only):Sets the precedence value in the tunnel header and the Frame Relay DE bit in a Frame Relay packet encapsulated for native L2TPv3 tunneling.
- **transmit**:Sends the packet with no alteration.

- **violate-action** *actions* : Actions taken on packets that exceed the PIR.
  - **drop**:Drops the packet.

You can configure these conform, exceed, and violate values for the *actions*argument of the **police** command in policy-map configuration mode on an ATM or Frame Relay ISE/E5 interface at the same time you use the **ip tos** command to configure the value of the ToS byte in IP headers of tunneled packets in a pseudowire class configuration applied to the interface (see the sections Configuring the L2TPv3 Pseudowire and *Manually Configuring L2TPv3 Session Parameters*).

However, the values you configure with the **police** command on an ISE/E5 interface for native L2TPv3 tunneling take precedence over any IP ToS configuration. This means that the traffic policing you configure always rewrites the IP header of the tunnel packet and overwrites the values set by an **ip tos** command. The priority of enforcement is as follows when you use these commands simultaneously:

1. **set-prec-tunnel** or **set-dscp-tunnel** (QoS policing in native L2TPv3 tunnel)
2. **ip tos reflect**
3. **ip tos tos-value**

Note This behavior is designed. We recommend that you configure only native L2TPv3 tunnel sessions and reconfigure any ISE/E5 interfaces configured with the **ip tos** command to use the QoS policy configured for native L2TPv3 traffic policing.

The following example shows how to configure traffic policing using the dual rate, 3-Color Marker on an ISE/E5 Frame Relay interface in a native L2TPv3 tunnel session.

Note This example shows how to use the **police**command in conjunction with the **conform-color** command to specify the policing actions to be taken on packets in the conform-color class and the exceed-color class. This is called a color-aware method of policing and is described in " QoS: Color-Aware Policer ." However, you can also configure color-blind traffic policing on an ISE/E5 Frame Relay interface in a native L2TPv3 tunnel session, using only the **police**command without the **conform-color** command.

```
class-map match-any match-not-frde
 match not fr-de
!
class-map match-any match-frde
 match fr-de
```

```
!
policy-map 2R3C_CA
 class class-default
  police cir 16000 bc 4470 pir 32000 be 4470
  conform-color match-not-frde exceed-color match-frde
  conform-action set-prec-tunnel-transmit 2
  exceed-action set-prec-tunnel-transmit 3
  exceed-action set-frde-transmit
  violate-action drop
```

The following example shows how to configure a QoS policy for traffic on the egress side of an ISE/E5 Frame Relay interface configured for a native L2TPv3 tunnel session.

Note that the sample output policy configured for a TSC-based L2TPv3 tunnel session in the section Configuring QoS on a Frame Relay Interface in a TSC-Based L2TPv3 Tunnel Session is not supported on a Frame Relay ISE/E5 interface. QoS policies on per-DLCI output traffic are not supported on ISE/E5 interfaces configured for a native L2TPv3 tunnel.

```
policy-map o
 class d2
  bandwidth percent 10
  random-detect precedence 1 200 packets 500 packets 1
 class d3
  bandwidth percent 10
  random-detect precedence 1 1 packets 2 packets 1
!
interface Serial0/1/1:0
 encapsulation frame-relay
 frame-relay interface-dlci 100 switched
  class fr-map
 service output o
```

Configuring Tunnel Marking in a Native L2TPv3 Tunnel Session

The QoS: Tunnel Marking for L2TPv3 Tunnels feature allows you to set (mark) either the IP precedence value or the differentiated services code point (DSCP) in the header of an L2TPv3 tunneled packet, using the **set-prec-tunnel** or **set-dscp-tunnel** command without configuring QoS traffic policing. Tunnel marking simplifies administrative overhead previously required to control customer bandwidth by allowing you to mark the L2TPv3 tunnel header on an ingress ISE/E5 interface.

The following example shows how to configure tunnel marking using MQC **set**commands for the default traffic class and a traffic class that matches a specified Frame Relay DE bit value:

```
class-map match-any match-frde
 match fr-de
policy-map set_prec_tun
 class match-frde
  set ip precedence tunnel 1
 class class-default
  set ip precedence tunnel 2
!
map-class frame-relay fr_100
 service-policy input set_prec_tun
```

L2TPv3 Customer-Facing ISE/E5 Interface

```
interface POS0/0
 frame-relay interface-dlci 100 switched
  class fr_100
```

Configuring Traffic Shaping in a Native L2TPv3 Tunnel Session

The following example shows how to configure traffic shaping on a Frame Relay ISE/E5 egress interface bound to a native L2TPv3 tunnel session. You can configure traffic shaping on a Frame Relay main egress interface by classifying traffic with different class maps.

Note You cannot configure per-DLCI shaping using the method shown in this example to configure traffic shaping.

To configure class-based shaping, configure the **match qos-group** and **random-detect discard-class** values according to the incoming IP precedence and DSCP values from packets received on the backbone-facing ingress interface. Use these values to define traffic classes on the customer-facing egress interface.

```
class-map match-any match_prec1
 match ip precedence 1
class-map match-any match_prec2
 match ip precedence 2
class-map match-any match_prec3
 match ip precedence 3
!
class-map match-all match_qos3
 match qos-group 3
!
class-map match-any match_qos12
 match qos-group 1
 match qos-group 2
!
policy-map customer_egress_policy
 class match_qos3
  bandwidth percent 5
  shape average 160000000
 class match_qos12
  shape average 64000000
  random-detect discard-class-based
  random-detect discard-class 1 500 packets 1000 packets
  random-detect discard-class 2 1000 packets 2000 packets
  bandwidth percent 10
 class class-default
  shape average 64000000
  queue-limit 1000 packets
  bandwidth percent 1
!
policy-map backbone_ingress_policy
 class match_prec1
  set qos-group 1
  set discard-class 1
 class match_prec2
  set qos-group 2
  set discard-class 2
 class match_prec3
  set qos-group 3
  set discard-class 3
 class class-default
  set qos-group 5
  set discard-class 5
```

L2TPv3 Customer-Facing ISE/E5 Interface

```
interface POS0/0
 service-policy output customer_egress_policy
 frame-relay interface-dlci 100 switched
  class fr_100
```

L2TPv3 Backbone-Facing ISE/E5 Interface

```
interface POS1/0
 service-policy input backbone_ingress_policy
```

Configuring a QoS Policy for Committed Information Rate Guarantees Example

The following example shows how to configure a QoS policy that guarantees a CIR of 256 kbps on DLCI 100 and 512 kbps for DLCI 200 on a serial interface at one end of a TSC-based L2TPv3 tunnel session:

```
ip cef distributed
 class-map dlci100
 match fr-dlci 100
 class-map dlci200
 match fr-dlci 200
!
policy-map dlci
 class dlci100
 bandwidth 256
 class dlci200
 bandwidth 512
!
interface Serial 0/0
 encapsulation frame-relay
 frame-relay intf-type dce
 service-policy output dlci
!
connect one Serial 0/0 100 l2transport
 xconnect 10.0.3.201 555 encapsulation l2tpv3 pw-class mqc
!
connect two Serial 0/0 200 l2transport
 xconnect 10.0.3.201 666 encapsulation l2tpv3 pw-class mqc
```

Setting the Frame Relay DE Bit Configuration Example

The following example shows how to configure the service policy called set-de and attach it to an output serial interface bound to a TSC-based L2TPv3 tunnel session. Note that setting the Frame Relay DE bit is not supported on a Frame Relay ISE/E5 interface bound to a native L2TPv3 tunnel session.

In this example, the class map called data evaluates all packets exiting the interface for an IP precedence value of 1. If the exiting packet has been marked with the IP precedence value of 1, the packet's DE bit is set to 1.

```
class-map data
 match qos-group 1
!
policy-map SET-DE
 class data
  set fr-de
!
interface Serial 0/0/0
 encapsulation frame-relay
 service-policy output SET-DE
!
connect fr-mpls-100 serial 0/0/0 100 l2transport
 xconnect 10.10.10.10 pw-class l2tpv3
```

Matching the Frame Relay DE Bit Configuration Example

The following example shows how to configure the service policy called match-de and attach it to an interface bound to a TSC-based L2TPv3 tunnel session. In this example, the class map called "data" evaluates all packets entering the interface for a DE bit setting of 1. If the entering packet has been a DE bit value of 1, the packet's IP precedence value is set to 3.

```
class-map data
 match fr-de
!
policy-map MATCH-DE
 class data
  set ip precedence tunnel 3
!
ip routing
ip cef distributed
!
mpls label protocol ldp
interface Loopback0
 ip address 10.20.20.20 255.255.255.255
!
interface Ethernet1/0/0
 ip address 172.16.0.2 255.255.255.0
 tag-switching ip
!
interface Serial4/0/0
 encapsulation frame-relay
service input MATCH-DE
!
connect 100 Serial4/0/0 100 l2transport
 xconnect 10.10.10.10 100 encapsulation l2tpv3
```

The next example shows how to configure the service policy called set_prec_tunnel_from_frde and attach it to a Cisco 12000 series ISE/E5 interface bound to a native L2TPv3 tunnel session. Note that in a native L2TPv3 session, you must attach the service policy to a DLCI (in the example, DCLI 100) instead of to a main interface (as in the preceding example).

```
class-map match-any match-frde
  match fr-de
!
policy-map set_prec_tunnel_from_frde
  class match-frde
   set ip precedence tunnel 6
  class class-default
   set ip precedence tunnel 3
!
map-class frame-relay fr_100
  service-policy input set_prec_tunnel_from_frde
!
interface POS0/0
  description ISE: L2TPv3 Customer-facing interface
  frame-relay interface-dlci 100 switched
    class fr_100
```

Configuring MLFR for L2TPv3 on the Cisco 12000 Series Example

The following example shows how to configure L2TPv3 tunneling on a multilink Frame Relay bundle interface on a Cisco 12000 series 2-port Channelized OC-3/STM-1 (DS1/E1) or 6-port Channelized T3 line card:

```
frame-relay switching
!
pseudowire-class mfr
```

```
 encapsulation l2tpv3
 ip local interface Loopback0
!
interface mfr0
 frame-relay intf-type dce
!
interface Serial0/0.1/1:11
 encapsulation frame-relay MFR0
!
interface Serial0/0.1/1:12
 encapsulation frame-relay MFR0
!
connect L2TPoMFR MFR0 100 l2transport
 xconnect 10.10.10.10 3 pw-class mfr
```

Configuring an MQC for Committed Information Rate Guarantees Example

The following is a sample configuration of the MQC to guarantee a CIR of 256 kbps on DLCI 100 and 512 kbps for DLCI 200:

```
ip cef distributed
 class-map dlci100
 match fr-dlci 100
 class-map dlci200
 match fr-dlci 200
!
policy-map dlci
 class dlci100
 bandwidth 256
 class dlci200
 bandwidth 512
!
interface Serial0/0
 encapsulation frame-relay
 frame-relay intf-type dce
 service-policy output dlci
!
connect one Serial0/0 100 l2transport
 xconnect 10.0.3.201 555 encapsulation l2tpv3 pw-class mqc
!
connect two Serial0/0 200 l2transport
 xconnect 10.0.3.201 666 encapsulation l2tpv3 pw-class mqc
```

Example: Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations

The following example shows how to configure an Ethertype other than 0x8100 on Gigabit Ethernet interfaces with QinQ or dot1Q encapsulations. In this example, the Ethertype field is set to 0x9100 on Gigabit Ethernet interface 1/0/0.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/0
Device(config-if)# dot1q tunneling ethertype 0x9100
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Wide area networking commands: complete command syntax, command mode, defaults, usage guidelines and examples	Cisco IOS Wide-Area Networking Command Reference
Configuring CEF	IP Switching Cisco Express Forwarding Configuration Guide
Frame Relay commands: complete command syntax, command mode, defaults, usage guidelines and examples	Cisco IOS Wide-Area Networking Command Reference
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines and examples	Cisco IOS IPv6 Command Reference
IPv6 configuration tasks	IPv6 Configuration Guide, Cisco IOS XE Release 3S
L2TP	• Layer 2 Tunnel Protocol • Layer 2 Tunneling Protocol: A Feature in Cisco IOS Software
L2TPv3	Layer 2 Tunneling Protocol Version 3 Technical Overview
L2VPN interworking	"L2VPN Interworking" chapter in the *MPLS Configuration Guide*
L2VPN pseudowire switching	"L2VPN Pseudowire Switching" chapter in the *MPLS Configuration Guide*
L2VPN pseudowire redundancy	"L2VPN Pseudowire Redundancy " chapter in the *Wide-Area Networking Configuration Guide*
MTU discovery and packet fragmentation	MTU Tuning for L2TP
Multilink Frame Relay over L2TPv3/AToM	Multilink Frame Relay over L2TPv3/AToM
Tunnel marking for L2TPv3 tunnels	QoS: Tunnel Marking for L2TPv3 Tunnels

Related Topic	Document Title
UTI	Universal Transport Interface (UTI)
VPN commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Dial Technologies Command Reference

Standards

Standard	Title
draft-ietf-l2tpext-l2tp-base-03.txt	Layer Two Tunneling Protocol (Version 3) "L2TPv3"

MIBs

MIB	MIBs Link
IfTable MIB for the attachment circuit	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2661	*Layer Two Tunneling Protocol "L2TP"*
RFC 1321	*The MD5 Message Digest Algorithm*
RFC 2104	*HMAC-Keyed Hashing for Message Authentication*
RFC 3931	*Layer Two Tunneling Protocol Version 3 "L2TPv3"*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Layer 2 Tunneling Protocol Version 3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for Layer 2 Tunneling Protocol Version 3

Release	Modification
2.6.2	Support was added for the **ip pmtu** command.
Cisco IOS Release 12.0	
12.0(21)S	Initial data plane support for L2TPv3 was introduced on the Cisco 7200 series, Cisco 7500 series, Cisco 10720, and Cisco 12000 series platforms.
12.0(23)S	L2TPv3 control plane support was introduced on the Cisco 7200 series, Cisco 7500 series, Cisco 10720, and Cisco 12000 series platforms.
12.0(24)S	L2TPv3 was enhanced to support the Layer 2 Fragmentation feature (fragmentation of IP packets before they enter the pseudowire) on the Cisco 7200 series, Cisco 7500 series, and Cisco 12000 series Internet routers.

Release	Modification
12.0(25)S	Support was added for the ATM VP Mode Single Cell Relay over L2TPv3 feature on the Cisco 7200 and Cisco 7500 series routers with ATM Deluxe PA-A3 interfaces. L2TPv3 control plane support was introduced on the Cisco 12000 series 1-port channelized OC-12 (DS3) line card.
12.0(23)S3	L2TPv3 control plane support was introduced on the Cisco 12000 series 1-port channelized OC-12 (DS3) line card.
12.0(24)S1	L2TPv3 control plane support was introduced on the Cisco 12000 series 1-port channelized OC-12 (DS3) line card.
12.0(27)S	Support was added for the following features to Cisco 12000 series 2-port channelized OC-3/STM-1 (DS1/E1) and 6-port Channelized T3 (T1) line cards: • Binding L2TPv3 sessions to Multilink Frame Relay (MLFR) interfaces • Quality of service (QoS) for Frame Relay attachment circuits
12.0(28)S	Support was added for the following features on the Cisco 7200 series and Cisco 7500 series routers: • ATM AAL5 OAM Emulation over L2TPv3 • ATM Single Cell Relay VC Mode over L2TPv3 • L2TPv3 Distributed Sequencing • L2TPv3 Support for PA-A3-8T1IMA PA and PA-A3-8E1IMA Port Adapters
12.0(29)S	Support was added for the following features: • ATM Cell Packing over L2TPv3 • ATM Port Mode Cell Relay over L2TPv3 • L2TPv3 Control Message Hashing • L2TPv3 Control Message Rate Limiting • Protocol Demultiplexing for L2TPv3

Release	Modification
12.0(30)S	Support was added for the following features to Cisco IOS Release 12.0(30)S: • L2TPv3 Digest Secret Graceful Switchover • Manual Clearing of L2TPv3 Tunnels • VC Class Provisioning for L2VPN Support was added for native L2TPv3 tunneling on IP services engine (ISE) line cards on the Cisco 12000 series Internet router.

12.0(31)S	Support was added for the following feature to Cisco IOS Release 12.0(31)S:
	• Layer 2 VPN (L2VPN): Syslog, SNMP Trap, and show Command Enhancements for AToM and L2TPv3
	
	Support was added for native L2TPv3 tunneling on the following ISE line cards on the Cisco 12000 series Internet router:
	• 2.5G ISE SPA Interface Processor (SIP):
	  • 2-port T3/E3 serial shared port adapter (SPA)
	  • 4-port T3/E3 serial SPA
	  • 2-port channelized T3 SPA
	  • 4-port channelized T3 Serial SPA
	• 4-port Gigabit Ethernet ISE

12.0(31)S2	Support was added for customer-facing IP Services Engine (ISE) interfaces configured for Layer 2 local switching on a Cisco 12000 series Internet router (see Layer 2 Local Switching).

Release	Modification
12.0(32)SY	Support was added for Engine 5 line cards--shared port adapters (SPAs) and SPA interface processors (SIPs)--on the Cisco 12000 series Internet router, including: • Engine-5 customer-facing interfaces that are configured for local switching (see Layer 2 Local Switching). • Engine-5 and ISE (Engine-3) interfaces that are configured for Layer 2 VPN interworking (see L 2VPN Interworking). Support was added for the L2TPv3 Layer 2 fragmentation feature on the Cisco 10720 Internet router.
12.0(33)S	Support was added for the following features to Cisco IOS Release 12.0(33)S: • Protocol Demultiplexing for L2TPv3 for PPP traffic • Protocol Demultiplexing for L2TPv3 for HDLC traffic • Protocol Demultiplexing for L2TPv3 on Engine-3/Engine-5 line cards in the Cisco 12000 series platforms • Protocol Demultiplexing for L2TPv3 on Engine-3/Engine-5 line cards in the Cisco 12000 series platforms for PPP, HDLC, Ethernet, and Frame-Relay encapsulations • Color Aware Policer on Engine-3/Engine-5 line cards for Ethernet over L2TPv3 • Site of Origin for Border Gateway Protocol Virtual Private Networks (BGP-VPNs) • Control Message Statistics and Conditional Debugging Command Enhancements (including L2VPN Pseudowire Conditional Debugging)
Cisco IOS Release 12.2S	

Release	Modification
12.2(25)S	Support was added for the following features to Cisco IOS Release 12.2(25)S: • L2TPv3: Layer 2 Tunneling Protocol • ATM AAL5 OAM Emulation over L2TPv3 • ATM Single Cell Relay VC Mode over L2TPv3 • ATM VP Mode Single Cell Relay over L2TPv3 • L2TPv3 Distributed Sequencing • L2TPv3 Layer 2 fragmentation • L2TPv3 Support for PA-A3-8T1IMA PA and PA-A3-8E1IMA Port Adapters
12.2(25)S4	Support was added for the following features on the Cisco 7304 NPE-G100 and the Cisco 7304 NSE-100: • L2TPv3: Layer 2 Tunneling Protocol • ATM AAL5 OAM Emulation over L2TPv3 • ATM Port Mode Cell Relay over L2TPv3 • ATM Single Cell Relay VC Mode over L2TPv3 • ATM VP Mode Single Cell Relay over L2TPv3 • L2TPv3 Layer 2 fragmentation Support was added for this feature on the Cisco 7304 NPE-G100 only: • L2TPv3 Distributed Sequencing
Cisco IOS Release 12.2SB	
12.2(27)SBC	Support was added for the following features: • L2TPv3 Control Message Hashing • L2TPv3 Control Message Rate Limiting • Layer 2 VPN (L2VPN): Syslog, SNMP Trap, and show Command Enhancements for AToM and L2TPv3 • Protocol Demultiplexing for L2TPv3

Release	Modification
12.2(28)SB	Support was added for Control Message Statistics and Conditional Debugging Command Enhancements (including L2VPN Pseudowire Conditional Debugging)
Cisco IOS Release 12.2SR	
12.2(33)SRC	The L2TPv3 feature was integrated into Cisco IOS Release 12.2(33)SRC and implemented on the Cisco 7600 series SPA Interface Processor-400 (SIP-400) line card.
Cisco IOS Release 12.3T	
12.3(2)T	The L2TPv3 feature was integrated into Cisco IOS Release 12.3(2)T and implemented on the Cisco 2600XM series Multiservice platforms, the Cisco 2691 Multiservice routers, the Cisco 3662 Multiservice Access platforms, the Cisco 3725 Modular Access routers, and the Cisco 3745 Modular Access routers.
Cisco IOS Release 12.4T	
12.4(11)T	Support was added for the following features: • L2TPv3 Control Message Hashing • L2TPv3 Control Message Rate Limiting • Protocol Demultiplexing for L2TPv3
Cisco IOS Release 15.0S	
15.0(1)S	Support was added for the following features: • ATM AAL5 OAM Emulation over L2TPv3 • ATM Single Cell Relay VC Mode over L2TPv3 • ATM VP Mode Single Cell Relay over L2TPv3 The following commands were introduced or modified: **atm mcpt-timers**, **atm pvp**, **cell-packing**, **clear l2tun**, **clear l2tun counters**, **clear l2tun counters tunnel l2tp**, **debug atm cell-packing**, **debug condition xconnect**, **debug vpdn**, **ip pmtu**, **i l2tp cookie local**, **l2tp cookie remote**, **l2tp hello**, **l2tp id**, and **xconnect**.

Glossary

AV pairs—Attribute-value pairs.

CEF—Cisco Express Forwarding. The Layer 3 IP switching technology that optimizes network performance and scalability for networks with large and dynamic traffic patterns.

data-link control layer—Layer 2 in the SNA architectural model. Responsible for the transmission of data over a particular physical link. Corresponds approximately to the data link layer of the OSI model.

DCE—Data circuit-terminating equipment (ITU-T expansion). Devices and connections of a communications network that comprise the network end of the user-to-network interface.

DF bit—Don't Fragment bit. The bit in the IP header that can be set to indicate that the packet should not be fragmented.

DTE—Data terminal equipment. The device at the user end of a user-network interface that serves as a data source, destination, or both.

HDLC—High-Level Data Link Control. A generic link-level communications protocol developed by the ISO. HDLC manages synchronous, code-transparent, serial information transfer over a link connection.

ICMP—Internet Control Message Protocol. A network protocol that handles network errors and error messages.

IDB— Interface descriptor block.

IS-IS—Intermediate System-to-Intermediate System. The OSI link-state hierarchical routing protocol based on DECnet Phase V routing, whereby ISs (devices) exchange routing information based on a single metric to determine network topology.

L2TP—An extension to PPP that merges features of two tunneling protocols: Layer 2 Forwarding (L2F) from Cisco Systems and Point-to-Point Tunneling Protocol (PPTP) from Microsoft. L2TP is an IETF standard endorsed by Cisco Systems and other networking industry leaders.

L2TPv3—The draft version of L2TP that enhances functionality in RFC 2661 (L2TP).

LMI—Local Management Interface.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the devices in the network where to forward packets based on preestablished IP routing information.

MQC—Modular quality of service CLI.

MTU—Maximum Transmission Unit. The maximum packet size, in bytes, that a particular interface can handle.

PMTU—Path MTU.

PVC—Permanent virtual circuit. A virtual circuit that is permanently established. A Frame Relay logical link, whose endpoints and class of service are defined by network management. Analogous to an X.25 permanent virtual circuit, a PVC consists of the originating Frame Relay network element address, originating data-link control identifier, terminating Frame Relay network element address, and termination data-link control identifier. Originating refers to the access interface from which the PVC is initiated. Terminating refers to the access interface at which the PVC stops. Many data network customers require a PVC between two points. PVCs save the bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. Data terminating equipment with a need for continuous communication uses PVCs.

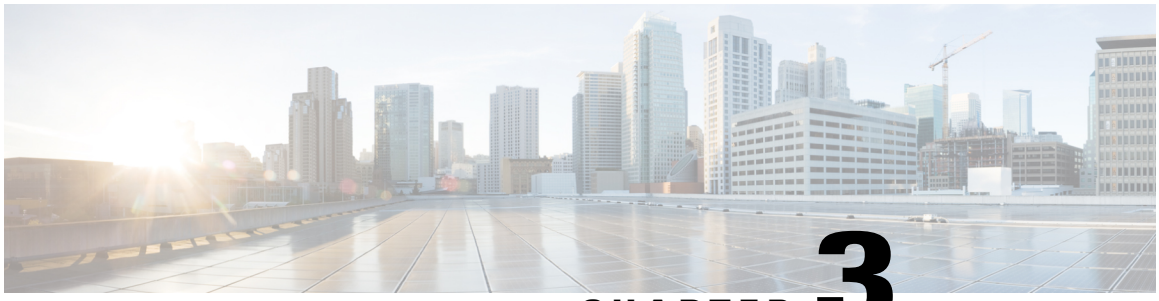
PW—Pseudowire.

SNMP—Simple Network Management Protocol. The network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and manage configurations, statistics collection, performance, and security.

tunneling—Architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

UNI—User-Network Interface.

VPDN—Virtual private dialup network. A network that allows separate and autonomous protocol domains to share common access infrastructure, including modems, access servers, and ISDN devices. A VPDN enables users to configure secure networks that take advantage of ISPs that tunnel remote access traffic through the ISP cloud.



CHAPTER 3

L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature lets you configure your network to detect a failure in the network and reroute the Layer 2 (L2) service to another endpoint that can continue to provide service. This feature provides the ability to recover from a failure either of the remote provider edge (PE) router or of the link between the PE and customer edge (CE) routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for L2VPN Pseudowire Redundancy

- This feature module requires that you understand how to configure basic L2 virtual private networks (VPNs). You can find that information in the following documents:
  - *Any Transport over MPLS*

- *L2 VPN Interworking*

- The L2VPN Pseudowire Redundancy feature requires that the following mechanisms be in place to enable you to detect a failure in the network:
  - Label-switched paths (LSP) Ping/Traceroute and Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV)
  - Local Management Interface (LMI)
  - Operation, Administration, and Maintenance (OAM)

Restrictions for L2VPN Pseudowire Redundancy

General Restrictions

- The primary and backup pseudowires must run the same type of transport service. The primary and backup pseudowires must be configured with AToM.
- Only static, on-box provisioning is supported.
- If you use L2VPN Pseudowire Redundancy with L2VPN Interworking, the interworking method must be the same for the primary and backup pseudowires.
- Setting the experimental (EXP) bit on the Multiprotocol Label Switching (MPLS) pseudowire is supported.
- Different pseudowire encapsulation types on the MPLS pseudowire are not supported.
- The **mpls l2transport route** command is not supported. Use the **xconnect** command instead.
- The ability to have the backup pseudowire fully operational at the same time that the primary pseudowire is operational is not supported. The backup pseudowire becomes active only after the primary pseudowire fails.
- The AToM VCCV feature is supported only on the active pseudowire.
- More than one backup pseudowire is not supported.
- Bidirectional Forwarding Detection over Virtual Circuit Connection Verification (BFDoVCCV) with status signaling is supported only on static pseudowires that do not have a backup peer. Explicit configuration of backup peers that violates this restriction is rejected.
- BFDoVCCV with status signaling through a pseudowire class is allowed. However, the feature is not supported on pseudowires that do not meet the restriction noted above.

Restrictions for Layer 2 Tunnel Protocol Version 3 (L2TPv3) Xconnect Configurations

- Interworking is not supported.
- Local switching backup by pseudowire redundancy is not supported.
- PPP, HDLC, and Frame-Relay attachment circuit (AC) types of L2TPv3 pseudowire redundancy are not supported.

- For the edge interface, only the Cisco 7600 series SPA Interface Processor-400 (SIP-400) linecard with the following shared port adapters (SPAs) is supported:

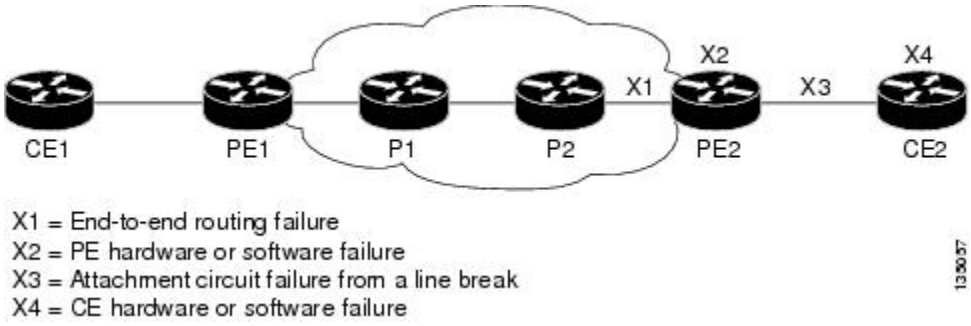
Cisco 2-Port Gigabit Ethernet Shared Port Adapter (SPA-2X1GE) Cisco 2-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-2X1GE-V2) Cisco 5-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-5X1GE-V2) Cisco 10-Port Gigabit Ethernet Shared Port Adapter, Version 2 (SPA-10X1GE-V2) Cisco 2-Port OC3c/STM1c ATM Shared Port Adapter (SPA-2XOC3-ATM) Cisco 4-Port OC3c/STM1c ATM Shared Port Adapter (SPA-4XOC3-ATM) Cisco 1-Port OC12c/STM4c ATM Shared Port Adapter (SPA-1XOC12-ATM) Cisco 1-Port OC-48c/STM-16 ATM Shared Port Adapter (SPA-1XOC48-ATM)

Information About L2VPN Pseudowire Redundancy

Introduction to L2VPN Pseudowire Redundancy

L2VPNs can provide pseudowire resiliency through their routing protocols. When connectivity between end-to-end PE devices fails, the L2VPN pseudowire redundancy can select and alternate path to the directed LDP session and the user data can take over. However, there are some parts of the network where this rerouting mechanism does not protect against interruptions in service. The figure below shows those parts of the network that are vulnerable to an interruption in service.

Figure 4: Points of Potential Failure in an L2VPN Network

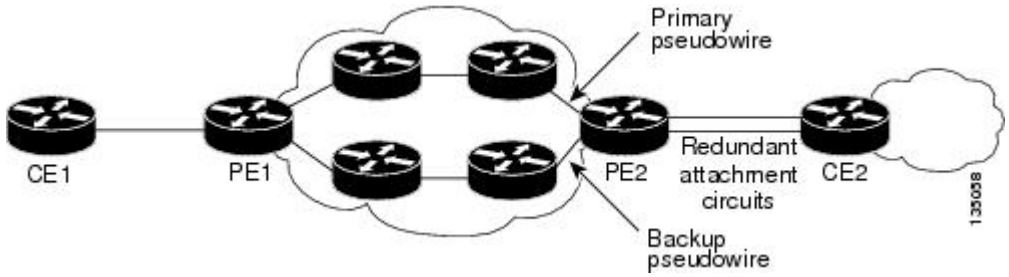


The L2VPN Pseudowire Redundancy feature provides the ability to ensure that the CE2 device in the figure above can always maintain network connectivity, even if one or all the failures in the figure occur.

The L2VPN Pseudowire Redundancy feature enables you to set up backup pseudowires. You can configure the network with redundant pseudowires (PWs) and redundant network elements, which are shown in the three figures below.

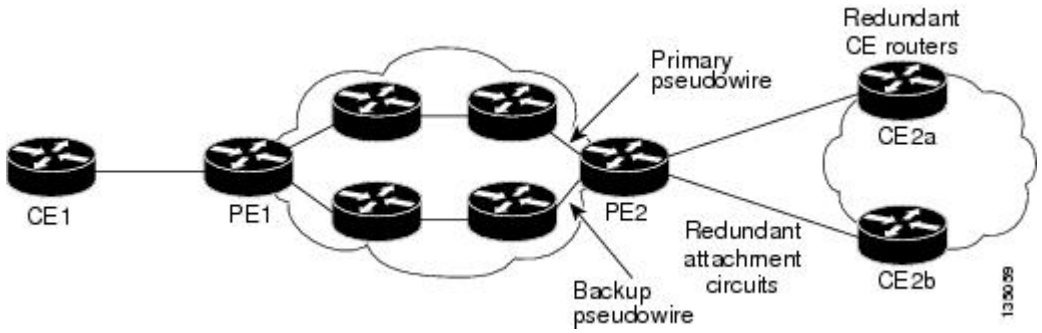
The figure below shows a network with redundant pseudowires and redundant attachment circuits.

Figure 5: L2VPN Network with Redundant PWs and Attachment Circuits



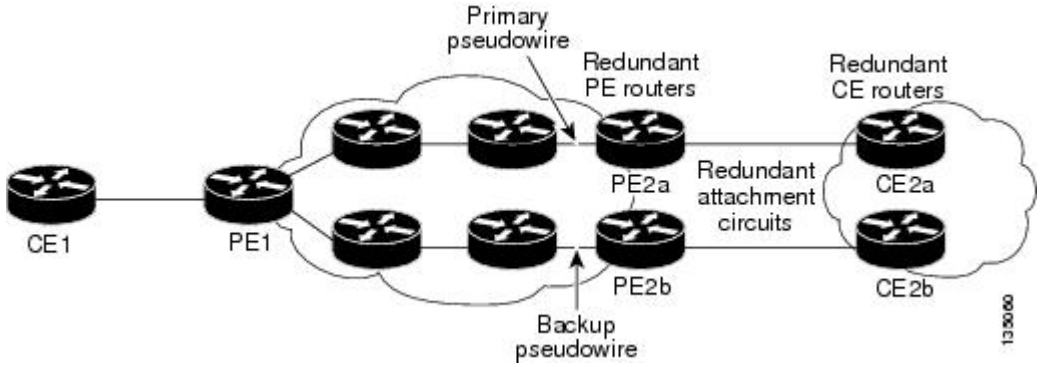
The figure below shows a network with redundant pseudowires, attachment circuits, and CE devices.

Figure 6: L2VPN Network with Redundant PWs, Attachment Circuits, and CE devices



The figure below shows a network with redundant pseudowires, attachment circuits, CE devices, and PE devices.

Figure 7: L2VPN Network with Redundant PWs, Attachment Circuits, CE devices, and PE devices



Xconnect as a Client of BFD

Redundant pseudowires are deployed to provide fault tolerance and resiliency to L2VPN-backhauled connections. The speed at which a system recovers from failures, especially when scaled to large numbers of

pseudowires, is critical to many service providers and service level agreements (SLAs). The configuration of a trigger for redundant pseudowire switchover reduces the time that it takes a large number of pseudowires to failover. A fundamental component of bidirectional forwarding detection (BFD) capability is enabled by fast-failure detection (FFD).

The configuration of this feature refers to a BFD configuration, such as the following (the second URL in the **bfd map** command is the loopback URL in the **monitor peer bfd** command):

```
bfd-template multi-hop mh
 interval min-tx 200 min-rx 200 multiplier 3 !
bfd map ipv4  10.1.1.0/24  10.1.1.1/32 mh
```

How to Configure L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to configure a backup pseudowire in case the primary pseudowire fails. When the primary pseudowire fails, the PE router can switch to the backup pseudowire. You can have the primary pseudowire resume operation after it comes back up.

The default Label Distribution Protocol (LDP) session hold-down timer will enable the software to detect failures in about 180 seconds. That time can be configured so that the software can detect failures more quickly. See the **mpls ldp holdtime** command for more information.

Configuring the Pseudowire

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.

The pseudowire-class configuration group specifies the characteristics of the tunneling mechanism, which are:

- Encapsulation type
- Control protocol
- Payload-specific options

You must specify the **encapsulation mpls**command as part of the pseudowire class for the AToM VCs to work properly. If you omit the **encapsulation mpls**command as part of the **xconnect**command, you receive the following error:

```
% Incomplete command.
```

Perform this task to configure a pseudowire class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** name
4. **encapsulation mpls**
5. **interworking** {**ethernet** | **ip**}

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	**pseudowire-class** name **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify. Enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. For AToM, the encapsulation type is **mpls**.
Step 5	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw-class)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuring L2VPN Pseudowire Redundancy

Use the following steps to configure the L2VPN Pseudowire Redundancy feature.

Before You Begin

For each transport type, the **xconnect**command is configured slightly differently. The following configuration steps use Ethernet VLAN over MPLS, which is configured in subinterface configuration mode. See *Any Transport over MPLS* to determine how to configure the **xconnect** command for other transport types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface*
4. **encapsulation dot1q** vlan-id
5. **xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}*
6. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*]
7. **backup delay** *enable-delay* {*disable-delay* | **never**}

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 gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface* **Example:** `Router(config)# interface gigabitethernet0/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**encapsulation dot1q** vlan-id **Example:** `Router(config-subif)# encapsulation dot1q 100`	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.
Step 5	**xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}* **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 pw-class atom`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. Enters xconnect configuration mode.
Step 6	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*]	Specifies a redundant peer for the pseudowire VC.

	Command or Action	Purpose
	Example: `Router(config-if-xconn)# backup peer 10.0.0.3 125 pw-class atom`	The pseudowire class name must match the name you specified when you created the pseudowire class, but you can use a different pw-class in the **backup peer** command than the name that you used in the primary **xconnect** command.
Step 7	**backup delay** *e nable-delay* {*disable-delay* | **never**} **Example:** `Router(config-if-xconn)# backup delay 5 never`	Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is 0 to 180. Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is 0 to 180 seconds. If you specify the **never keyword**, the primary pseudowire VC never takes over for the backup.

Configuring Xconnect as a Client of BFD

Perform this task to configure a trigger for redundant pseudowire switchover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class mpls-ffd**
   - Enters pseudowire class configuration mode.
4. **encapsulation mpls**
5. **monitor peer bfd** [**local interface** *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pseudowire-class mpls-ffd** • Enters pseudowire class configuration mode. **Example:** `Device(config)# pseudowire-class mpls-ffd`	Establishes a pseudowire class for MPLS fast-failure detection.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation to be MPLS.
Step 5	**monitor peer bfd** [**local interface** *interface-type interface-number*] **Example:** `Device(config-pw-class)# monitor peer bfd local interface loopback 0`	Enables the pseudowire fast-failure detection capability.

Forcing a Manual Switchover to the Backup Pseudowire VC

To force the router switch over to the backup or primary pseudowire, you can enter the **xconnect backup force switchover** command in privileged EXEC mode. You can specify either the interface of the primary attachment circuit (AC) to switch to or the IP-address and VC ID of the peer router.

A manual switchover can be made only if the interface or peer specified in the command is actually available and the xconnect will move to the fully active state when the command is entered.

SUMMARY STEPS

1. **enable**
2. **xconnect backup force-switchover {** **interface** *interface-info* | **peer** *ip-address vcid}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**xconnect backup force-switchover { interface** *interface-info* | **peer** *ip-address vcid}* **Example:** `Router# xconnect backup force-switchover peer 10.10.10.1 123`	Specifies that the router should switch to the backup or to the primary pseudowire.

Verifying the L2VPN Pseudowire Redundancy Configuration

Use the following commands to verify that the L2VPN Pseudowire Redundancy feature is correctly configured.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show xconnect all**
3. **xconnect logging redundancy**

DETAILED STEPS

Step 1 **show mpls l2transport vc**

In this example, the primary attachment circuit is up. The backup attachment circuit is available, but not currently selected. The **show** output displays as follows:

Example:

```
Router# show mpls l2transport vc
Local intf     Local circuit           Dest address    VC ID      Status
-------------  ----------------------- --------------- ---------- ----------
Et0/0.1        Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1        Eth VLAN 101            10.0.0.3        201        DOWN

Router# show mpls l2transport vc detail
Local interface: Et0/0.1 up, line protocol up, Eth VLAN 101 up
   Destination address 10.0.0.2 VC ID: 101, VC status UP
   .
   .
   .
Local interface: Et0/0.1 down, line protocol down, Eth VLAN 101 down
   Destination address 10.0.0.3 VC ID: 201, VC status down
   .
   .
   .
```

Step 2 **show xconnect all**

In this example, the topology is Attachment Circuit 1 to Pseudowire 1 with a Pseudowire 2 as a backup:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
------+------------------------------------+--+------------------------------------+--
UP pri ac   Et0/0(Ethernet)              UP mpls 10.55.55.2:1000              UP
IA sec ac   Et0/0(Ethernet)              UP mpls 10.55.55.3:1001              DN
```

In this example, the topology is Attachment Circuit 1 to Attachment Circuit 2 with a Pseudowire backup for Attachment Circuit 2:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
------+------------------------------------+--+------------------------------------+--
UP pri ac   Se6/0:150(FR DLCI)           UP ac   Se8/0:150(FR DLCI)           UP
IA sec ac   Se6/0:150(FR DLCI)           UP mpls 10.55.55.3:7151              DN
```

Step 3 **xconnect logging redundancy**

In addition to the **show mpls l2transport vc**command and the **show xconnect** command, you can use the **xconnect logging redundancy** command to track the status of the xconnect redundancy group:

Example:

```
Router(config)# xconnect logging redundancy
```

When this command is configured, the following messages will be generated during switchover events:

Activating the primary member:

Example:

```
00:01:07: %XCONNECT-5-REDUNDANCY: Activating primary member 10.55.55.2:1000
```

Activating the backup member:

Example:

```
00:01:05: %XCONNECT-5-REDUNDANCY: Activating secondary member 10.55.55.3:1001
```

Configuration Examples for L2VPN Pseudowire Redundancy

Each of the configuration examples refers to one of the following pseudowire classes:

- AToM (like-to-like) pseudowire class:

```
pseudowire-class mpls
 encapsulation mpls
```

- L2VPN IP interworking:

```
pseudowire-class mpls-ip
 encapsulation mpls
 interworking ip
```

L2VPN Pseudowire Redundancy and AToM Like to Like Examples

The following example shows a High-Level Data Link Control (HDLC) attachment circuit xconnect with a backup pseudowire:

```
interface Serial4/0
 xconnect 10.55.55.2 4000 pw-class mpls
 backup peer 10.55.55.3 4001 pw-class mpls
```

The following example shows a Frame Relay attachment circuit xconnect with a backup pseudowire:

```
connect fr-fr-pw Serial6/0 225 l2transport
 xconnect 10.55.55.2 5225 pw-class mpls
 backup peer 10.55.55.3 5226 pw-class mpls
```

L2VPN Pseudowire Redundancy and L2VPN Interworking Examples

The following example shows an Ethernet attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet0/0
 xconnect 10.55.55.2 1000 pw-class mpls-ip
 backup peer 10.55.55.3 1001 pw-class mpls-ip
```

The following example shows an Ethernet VLAN attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet1/0.1
 encapsulation dot1Q 200
 no ip directed-broadcast
 xconnect 10.55.55.2 5200 pw-class mpls-ip
 backup peer 10.55.55.3 5201 pw-class mpls-ip
```

The following example shows a Frame Relay attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
connect fr-ppp-pw Serial6/0 250 l2transport
 xconnect 10.55.55.2 8250 pw-class mpls-ip
 backup peer 10.55.55.3 8251 pw-class mpls-ip
```

The following example shows a PPP attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Serial7/0
 encapsulation ppp
 xconnect 10.55.55.2 2175 pw-class mpls-ip
 backup peer 10.55.55.3 2176 pw-class mpls-ip
```

L2VPN Pseudowire Redundancy with Layer 2 Local Switching Examples

The following example shows an Ethernet VLAN-VLAN local switching xconnect with a pseudowire backup for Ethernet segment E2/0.2. If the subinterface associated with E2/0.2 goes down, the backup pseudowire is activated.

```
connect vlan-vlan Ethernet1/0.2 Ethernet2/0.2
 backup peer 10.55.55.3 1101 pw-class mpls
```

The following example shows a Frame Relay-to-Frame Relay local switching connect with a pseudowire backup for Frame Relay segment S8/0 150. If data-link connection identifier (DLCI) 150 on S8/0 goes down, the backup pseudowire is activated.

```
connect fr-fr-ls Serial6/0 150 Serial8/0 150
 backup peer 10.55.55.3 7151 pw-class mpls
```

Additional References

Related Documents

Related Topic	Document Title
Any Transport over MPLS	Any Transport over MPLS
High Availability for AToM	AToM Graceful Restart
L2VPN Interworking	L2VPN Interworking
Layer 2 local switching	Layer 2 Local Switching
PWE3 MIB	Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services
Packet sequencing	Any Transport over MPLS (AToM) Sequencing Support
BFD configuration	IP Routing BFD Configuration Guide

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for L2VPN Pseudowire Redundancy

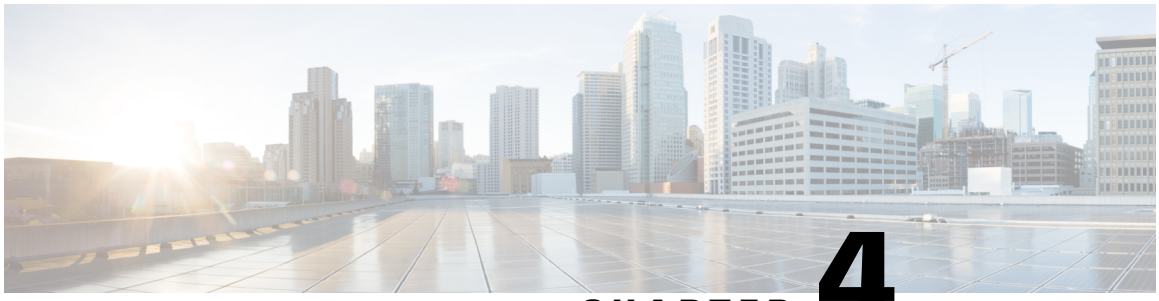
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for L2VPN Pseudowire Redundancy

Feature Name	Releases	Feature Information
L2VPN Pseudowire Redundancy	12.0(31)S 12.2(28)SB 12.2(22)SXI 12.2(33)SRB 12.4(11)T 15.0(1)S	This feature enables you to set up your network to detect a failure in the network and reroute the Layer 2 service to another endpoint that can continue to provide service. In Cisco IOS Release 12.0(31)S, the L2VPN Pseudowire Redundancy feature was introduced for Any Transport over MPLS (AToM) on the Cisco 12000 series routers. This feature was integrated into Cisco IOS Release 12.2(28)SB. This feature was integrated into Cisco IOS Release 12.4(11)T. This feature was integrated into Cisco IOS Release 12.2(33)SRB. This feature was integrated into Cisco IOS Release 12.2(33)SXI. The following commands were introduced or modified: **backup delay (L2VPN local switching)**, **backup peer**, **show xconnect**, **xconnect backup force-switchover**, **xconnect logging redundancy**.
L2VPN Pseudowire Redundancy for L2TPv3	12.2(33)SRE 15.0(1)S	This feature provides L2VPN pseudowire redundancy for L2TPv3 xconnect configurations. In Cisco IOS Release 12.2(33)SRE, this feature was implemented on the Cisco 7600 series routers.
Xconnect as a Client of BFD	15.1(3)S	This feature provides fast-failure detection for L2VPN pseudowire redundancy. The following command was introduced: **monitor peer bfd**.

Feature Name	Releases	Feature Information
Resilient Pseudowire (RPW): PW Fast Recovery	15.2(1)S	This feature was integrated into Cisco IOS Release 15.2(1)S. The following commands were introduced or modified: **aps hspw-icrm-grp** , **show hspw-aps-icrm**.



CHAPTER 4

L2VPN Interworking

Layer 2 VPN (L2VPN) Interworking allows you to connect disparate attachment circuits. This feature module explains how to configure the following L2VPN Interworking features:

- Ethernet/VLAN to ATM adaptation layer 5 (AAL5) Interworking
- Ethernet/VLAN to Frame Relay Interworking
- Ethernet/VLAN to PPP Interworking
- Ethernet to VLAN Interworking
- Frame Relay to ATM AAL5 Interworking
- Frame Relay to PPP Interworking
- Ethernet/VLAN to ATM virtual path identifier (VPI) and virtual channel identifier (VCI) Interworking
- L2VPN Interworking: VLAN Enable/Disable Option for Any Transport over MPLS (AToM)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for L2VPN Interworking

Before you configure L2VPN Interworking on a device, do the following:

- You must enable Cisco Express Forwarding.
- On Cisco 12000 series Internet routers, before you configure Layer 2 Tunnel Protocol Version 3 (L2TPv3) for L2VPN Interworking on an IP Services Engine (ISE), such as an Engine 3 or Engine 5 interface, you must enable the L2VPN feature bundle on the line card.
  To enable the feature bundle, enter the **hw-module slot np mode feature** command in global configuration mode as follows:

```
Device# configure terminal
Device(config)# hw-module slot slot-number np mode feature
```

Restrictions for L2VPN Interworking

General Restrictions

This section lists the general restrictions that apply to L2VPN Interworking. Other restrictions that are platform-specific or device-specific are listed in the subsequent sections.

- The interworking type on one provider edge (PE) router must match the interworking type on the peer PE router.
- The following quality of service (QoS) features are supported with L2VPN Interworking:
  - Static IP type of service (ToS) or Multiprotocol Label Switching (MPLS) experimental (EXP) bit setting in tunnel header
  - IP ToS reflection in the tunnel header (Layer 2 Tunnel Protocol Version 3 [L2TPv3] only)
  - Frame Relay policing
  - Frame Relay data-link connection identifier (DLCI)-based congestion management (Cisco 7500/Versatile Interface Processor [VIP])
  - One-to-one mapping of VLAN priority bits to MPLS EXP bits
- Only ATM AAL5 virtual circuit (VC) mode is supported. ATM virtual path (VP) and port mode are not supported.
- In Cisco IOS Release 12.2(52)SE and Cisco IOS Release 12.2(33)SRE, the **encapsulation** command supports only the **mpls** keyword. The **l2tpv3** keyword is not supported. The **interworking** command supports only the **ethernet** and **vlan** keywords. The **ip** keyword is not supported.

Cisco 7600 Series Routers Restrictions

The table below lists the line cards that are supported on Cisco 7600 series routers. These line cards are supported on the WAN (ATM, Frame Relay, or PPP) side of the interworking link. The second table below shows the line cards that are supported on the Ethernet side of the interworking link. For more details about shared port adapters and line cards supported on Cisco 7600 series routers, see the *Cross-Platform Release Notes for Cisco IOS Release 12.2SR* document for Cisco 7600 series routers.

Table 11: Cisco 7600 Series Routers: Supported Line Cards for the WAN Side

Interworking Type	Core-Facing Line Cards	Customer-Edge Line Cards
Ethernet (bridged) (ATM and Frame Relay)	Any	EflexWAN SIP-200 SIP-400
IP (routed) (ATM, Frame Relay, and PPP)	Any	EflexWAN SIP-200

Table 12: Cisco 7600 Series Routers: Supported Line Cards for the Ethernet Side

Interworking Type	Ethernet over MPLS Mode	Core-Facing Line Cards	Customer-Edge Line Cards
Ethernet (bridged)	Policy feature card (PFC)-based	Any, except optical service module (OSM) and ES40	Catalyst LAN SIP-600
	Switched virtual interface (SVI)-based	EflexWAN ES20 ES+40 SIP-200 SIP-400 SIP-600	Catalyst LAN EflexWAN (with Multipoint Bridging [MPB]) ES20 ES+40 SIP-200 (with MPB) SIP-400 (with MPB) SIP-600
	Scalable (with E-MPB)	Any, except OSM	ES20 SIP-600 and SIP-400 with Gigabit Ethernet (GE) shared port adapter (SPA)

Interworking Type	Ethernet over MPLS Mode	Core-Facing Line Cards	Customer-Edge Line Cards
IP (routed)	PFC-based	Catalyst LAN SIP-600 **Note:** PFC-based mode is not supported with routed interworking in Cisco IOS Release 12.2(33)SRD. Use SVI, Scalable, or Ethernet virtual connection (EVC)-based Ethernet over MPLS (EoMPLS) instead.	Catalyst LAN SIP-600 **Note:** PFC-based mode is not supported with routed interworking in Cisco IOS Release 12.2(33)SRD. Use SVI, Scalable, or EVC-based EoMPLS instead.
	SVI-based	Any, except Catalyst LAN and OSM.	Catalyst LAN EflexWAN (with MPB) ES20 SIP-200 (with MPB) SIP-400 (with MPB) SIP-600

The following restrictions apply to Cisco 7600 series routers and L2VPN Interworking:

- Operation, Administration, and Management (OAM) emulation is not required with L2VPN Interworking on the SIP-200, SIP-400, and Flexwan2 line cards.
- Cisco 7600 series routers support the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature starting in Cisco IOS Release 12.2(33)SRE. This feature has the following restrictions:
  - PFC-based EoMPLS is not supported.
  - Scalable and SVI-based EoMPLS are supported with the SIP-400 line card.
- Cisco 7600 series routers do not support L2VPN Interworking over L2TPv3.
- Cisco 7600 series routers support only the following interworking types:
  - Ethernet/VLAN to Frame Relay (IP and Ethernet modes)
  - Ethernet/VLAN to ATM AAL5SNAP (IP and Ethernet modes)
  - Ethernet/VLAN to PPP (IP only)
  - Ethernet to VLAN Interworking
- Cisco 7600 series routers do not support the following interworking types:
  - Ethernet/VLAN to ATM AAL5MUX
  - Frame Relay to PPP Interworking

    - Frame Relay to ATM AAL5 Interworking

- Both ends of the interworking link must be configured with the same encapsulation and interworking type, as described below:
    - If you use Ethernet encapsulation, you must use the Ethernet (bridged) interworking type. If you are not using Ethernet encapsulation, you can use a bridging mechanism such as routed bridge encapsulation (RBE).
    - If you use an IP encapsulation (such as ATM or Frame Relay), you must use the IP (routed) interworking type. The PE routers negotiate the process for learning and resolving addresses.
    - You must use the same maximum transmission unit (MTU) size on the attachment circuits at each end of the pseudowire.
- Frame Relay interworking does not pass the IEEE spanning tree bridge protocol data units (BPDUs) when Cisco 7600 series routers are used as PEs.
- PFC-based EoMPLS is not supported on ES40 line cards. SVI and EVC/scalable EoMPLS are the alternative options.
- PFC-based EoMPLS is not supported for routed/IP interworking in Cisco IOS Release 12.2(33)SRD and later releases. The alternative routed/IP interworking options are SVI and EVC or scalable EoMPLS. However, PFC-based EoMPLS is supported for Ethernet/Bridged interworking and for like-to-like over AToM.

Cisco 12000 Series Internet Routers Restrictions

For more information about hardware requirements on Cisco 12000 series Internet routers, see the *Cross-Platform Release Notes for Cisco IOS Release 12.0S* document.

For quality of service (QoS) support on Cisco 12000 series Internet routers, see the *Any Transport over MPLS (AToM): Layer 2 QoS for the Cisco 12000 Series Router (Quality of Service)* configuration guide.

Frame Relay to PPP and High-Level Data Link Control Interworking

Cisco 12000 series Internet routers do not support L2VPN Interworking with PPP and high-level data link control (HDLC) transport types in Cisco IOS releases earlier than Cisco IOS Release 12.0(32)S.

In Cisco IOS Release 12.0(32)S and later releases, Cisco 12000 series Internet routers support L2VPN interworking for Frame Relay over MPLS and PPP and HDLC over MPLS only on the following shared port adapters (SPAs):

- ISE/Engine 3 SPAs:
    - SPA-2XCT3/DS0 (2-port channelized T3 to DS0)
    - SPA-4XCT3/DS0 (4-port channelized T3 to DS0)
- Engine 5 SPAs:
    - SPA-1XCHSTM1/OC-3 (1-port channelized STM-1c/OC-3c to DS0)
    - SPA-2XOC-48-POS/RPR (2-port OC-48/STM16 POS/RPR)

- SPA-8XCHT1/E1 (8-port channelized T1/E1)
- SPA-OC-192POS-LR (1-port OC-192/STM64 POS/RPR)
- SPA-OC-192POS-XFP (1-port OC-192/STM64 POS/RPR)

L2VPN Interworking over L2TPv3

On Cisco 12000 series Internet routers, Ethernet (bridged) interworking is not supported for L2TPv3. Only IP (routed) interworking is supported.

IP (routed) interworking is not supported in an L2TPv3 pseudowire that is configured for data sequencing (using the **sequencing** command).

In Cisco IOS Release 12.0(32)SY and later releases, Cisco 12000 series Internet routers support L2VPN Interworking over L2TPv3 tunnels in IP mode on ISE and Engine 5 line cards as follows:

- On an ISE interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported:
  - 802.1q (VLAN)
  - ATM adaptation layer type-5 (AAL5)
  - Ethernet
  - Frame Relay DLCI
- On an Engine 5 interface configured for L2TPv3 tunneling, the following Layer 2 encapsulations are supported:
  - 802.1q (VLAN)
  - Ethernet
  - Frame Relay DLCI

For more information, refer to the “Layer 2 Tunneling Protocol Version 3” module.

The only frame format supported for L2TPv3 interworking on Engine 5 Ethernet SPAs is Ethernet Version 2 (also known as Ethernet II) with the Ether type 0x0800 value set as Internet Protocol Payload and optionally, 802.1q VLAN. Ethernet packets with other Ethernet frame formats are dropped.

Remote Ethernet Port Shutdown Support

The Cisco Remote Ethernet Port Shutdown feature (which minimizes potential data loss after a remote link failure) is supported only on the following Engine 5 Ethernet SPAs:

- SPA-8XFE (8-port Fast Ethernet)
- SPA-2X1GE (2-port Gigabit Ethernet)
- SPA-5X1GE (5-port Gigabit Ethernet)
- SPA-10X1GE (10-port Gigabit Ethernet)
- SPA-1X10GE (1-port 10-Gigabit Ethernet)

For more information about this feature, refer to the *Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown* configuration guide.

L2VPN Any-to-Any Interworking on Engine 5 Line Cards

For more information about hardware requirements on Cisco 12000 series Internet routers, see the *Cross-Platform Release Notes for Cisco IOS Release 12.0S* document.

For quality of service (QoS) support on Cisco 12000 series Internet routers, see the *Any Transport over MPLS (AToM): Layer 2 QoS for the Cisco 12000 Series Router (Quality of Service)* configuration guide.

L2VPN Any-to-Any Interworking on Engine 5 Line Cards

The table below shows the different combinations of transport types supported for L2VPN Interworking on Engine 3 and Engine 5 SPA interfaces connected through an attachment circuit over MPLS or L2TPv3.

Table 13: Engine 3 and Engine 5 Line Cards/SPAs Supported for L2VPN Interworking

Attachment Circuit 1 (AC1)	Attachment Circuit 2 (AC2)	Interworking Mode	AC1 Engine Type and Line Card/SPA	AC2 Engine Type and Line Card/SPA
Frame Relay	Frame Relay	IP	Engine 5 Packet over SONET (POS) and channelized	Engine 3 ATM line cards
Frame Relay	ATM	Ethernet	Engine 5 POS and channelized	Engine 3 ATM line cards
Frame Relay	ATM	IP	Engine 5 POS and channelized	Engine 3 ATM line cards
Frame Relay	Ethernet	Ethernet	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	Ethernet	IP	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	VLAN	Ethernet	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Frame Relay	VLAN	IP	Engine 5 POS and channelized	Engine 5 Gigabit Ethernet
Ethernet	Ethernet	Ethernet	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
Ethernet	Ethernet	IP	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
Ethernet	VLAN	Ethernet	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet

Attachment Circuit 1 (AC1)	Attachment Circuit 2 (AC2)	Interworking Mode	AC1 Engine Type and Line Card/SPA	AC2 Engine Type and Line Card/SPA
Ethernet	VLAN	IP	Engine 5 Gigabit Ethernet	Engine 5 Gigabit Ethernet
ATM	Ethernet	Ethernet	Engine 3 ATM line cards	Engine 5 Gigabit Ethernet
ATM	Ethernet	IP	Engine 3 ATM line cards	Engine 5 Gigabit Ethernet

On the Cisco 12000 series Engine 3 line card, Network Layer Protocol ID (NLPID) encapsulation is not supported in routed mode; neither NLPID nor AAL5MUX is supported in bridged mode. On Cisco 12000 series Internet routers, Ethernet (bridged) interworking is not supported for L2TPv3.

In an L2VPN Interworking configuration, after you configure L2TPv3 tunnel encapsulation for a pseudowire using the **encapsulation l2tpv3** command, you cannot enter the **interworking ethernet** command.

On Ethernet SPAs on Cisco 12000 series Internet routers, the only frame format supported for L2TPv3 interworking is Ethernet Version 2 (also known as Ethernet II) with the Ether type 0x0800 value set as Internet Protocol Payload and optionally, 802.1q VLAN. Ethernet packets with other Ethernet frame formats are dropped.

ATM AAL5 Interworking Restrictions

The following restrictions apply to ATM AAL5 Interworking:

- Switched virtual circuits (SVCs) are not supported.
- Inverse Address Resolution Protocol (ARP) is not supported with IP interworking.
- Customer edge (CE) routers must use point-to-point subinterfaces or static maps.
- In the Ethernet end-to-end over ATM scenario, the translations listed below are supported. Everything else is dropped.
    - Ethernet without LAN frame check sequence (FCS) (AAAA030080C200070000)
    - Spanning tree (AAAA030080c2000E)
- In the IP over ATM scenario, the IPv4 (AAAA030000000800) translation is supported. Everything else is dropped.
- Operation, Administration, and Management (OAM) emulation for L2VPN Interworking is the same as like-to-like. The end-to-end F5 loopback cells are looped back on the PE router. When the pseudowire is down, an F5 end-to-end segment Alarm Indication Signal (AIS)/Remote Defect Identification (RDI) is sent from the PE router to the CE router.
- Interim Local Management Interface (ILMI) can manage virtual circuits (VCs) and permanent virtual circuits (PVCs).

- To enable ILMI management, configure ILMI PVC 0/16 on the PE router's ATM interface. If a PVC is provisioned or deleted, an ilmiVCCChange trap is sent to the CE router.
- Only the user side of the User-Network Interface (UNI) is supported; the network side of the UNI is not supported.

Ethernet VLAN Interworking Restrictions

The following restrictions apply to Ethernet/VLAN interworking:

- When you configure VLAN to Ethernet interworking, VLAN to Frame Relay (routed) interworking, or ATM using Ethernet (bridged) interworking, the PE router on the Ethernet side that receives a VLAN tagged frame from the CE router removes the VLAN tag. In the reverse direction, the PE router adds the VLAN tag to the frame before sending the frame to the CE router.
  (If you enable the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature with the **interworking vlan** command, the VLAN ID is included as part of the Ethernet frame. See the "VLAN Interworking" section for more information.)
- In bridged interworking from VLAN to Frame Relay, the Frame Relay PE router does not strip off VLAN tags from the Ethernet traffic it receives.
- Cisco 10720 series Internet routers support Ethernet to VLAN Interworking Ethernet only over L2TPv3.
- Ethernet interworking for a raw Ethernet port or a VLAN trunk is not supported. Traffic streams are not kept separate when traffic is sent between transport types.
- In routed mode, only one CE router can be attached to an Ethernet PE router.
- There must be a one-to-one relationship between an attachment circuit and the pseudowire. Point-to-multipoint or multipoint-to-point configurations are not supported.
- Ensure that you configure routing protocols for point-to-point operation on the CE routers when configure Ethernet to non-Ethernet.
- In the IP interworking mode, the IPv4 (0800) translation is supported. The PE router captures ARP (0806) packets and responds with its own MAC address (proxy ARP). Everything else is dropped.
- The Ethernet or VLAN must contain only two IP devices: a PE router and a CE router. The PE router performs proxy ARP and responds to all ARP requests it receives. Therefore, only one CE and one PE router should be on the Ethernet or VLAN segment.
- If CE routers are configured for static routing, perform the following tasks:
  - The PE router needs to learn the MAC address of the CE router to correctly forward traffic to it. The Ethernet PE router sends an Internet Control Message Protocol (ICMP) Router discovery protocol (RDP) solicitation message with the source IP address as zero. The Ethernet CE router responds to this solicitation message. To configure Cisco CE router's Ethernet or VLAN interface to respond to the ICMP RDP solicitation message, use the **ip irdp** command in interface configuration mode. If you do not configure the CE router, traffic is dropped until the CE router sends traffic towards the PE router.
  - To disable the CE routers from running the router discovery protocol, issue the **ip irdp maxadvertinterval 0** command in interface mode.

- Ethernet interworking between an Ethernet port and a VLAN supports spanning tree protocol only on VLAN 1. Configure VLAN 1 as a nonnative VLAN. This restriction applies if you configure interworking between Ethernet and VLAN with Catalyst switches as the CE routers. The spanning tree protocol is supported for Ethernet interworking.
- When you change the interworking configuration on an Ethernet PE router, clear the ARP entry on the adjacent CE router so that it can learn the new MAC address. Otherwise, you might experience traffic drops.

L2VPN Interworking VLAN Enable/Disable Option for AToM Restrictions

The following restrictions apply to the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature, which allows the VLAN ID to be included as part of the Ethernet frame:

- The L2VPN Interworking: VLAN Enable/Disable Option for AToM feature is supported in the following releases:
  - Cisco IOS Release 12.2(52)SE for Cisco Catalyst 3750 Metro switches
  - Cisco IOS Release 12.2(33)SRE for Cisco 7600 series routers
- The L2VPN Interworking: VLAN Enable/Disable Option for AToM feature is not supported with L2TPv3. You can configure the feature only with AToM.
- If the interface on the PE router is a VLAN interface, the **interworking vlan** command need not be specified on that PE router.
- The L2VPN Interworking: VLAN Enable/Disable Option for AToM feature works only with the following attachment circuit combinations:
  - Ethernet to Ethernet
  - Ethernet to VLAN
  - VLAN to VLAN
- If you specify an interworking type on a PE router, that interworking type must be enforced. The interworking type must match on both PE routers. Otherwise, the VC may be in an incompatible state and remain in the down state. If the attachment circuit (AC) is VLAN, the PE router can negotiate (autosense) the VC type using the Label Distribution Protocol (LDP).
  For example, both PE1 and PE2 use Ethernet interfaces, and VLAN interworking is specified on PE1 only. PE2 is not configured with an interworking type and cannot autosense the interworking type. The result is an incompatible state where the VC remains in the down state.

  However, if PE1 uses an Ethernet interface and VLAN interworking is enabled (which will enforce VLAN as the VC type), and PE2 uses a VLAN interface and interworking is not enabled (which causes PE2 to use Ethernet as its default VC type), PE2 can autosense and negotiate the interworking type and select VLAN as the VC type.

The table below shows the AC types, interworking options, and VC types after negotiation.

Table 14: Negotiating Ethernet and VLAN Interworking Types

PE1 AC Type	Interworking Option	PE2 AC Type	Interworking Option	VC Type after Negotiation
Ethernet	None	Ethernet	None	Ethernet
VLAN	None	Ethernet	None	Ethernet
Ethernet	None	VLAN	None	Ethernet
VLAN	None	VLAN	None	Ethernet
Ethernet	VLAN	Ethernet	None	Incompatible
VLAN	VLAN	Ethernet	None	Incompatible
Ethernet	VLAN	VLAN	None	VLAN
VLAN	VLAN	VLAN	None	VLAN
Ethernet	None	Ethernet	VLAN	Incompatible
VLAN	None	Ethernet	VLAN	VLAN
Ethernet	None	VLAN	VLAN	Incompatible
VLAN	None	VLAN	VLAN	VLAN
Ethernet	VLAN	Ethernet	VLAN	VLAN
VLAN	VLAN	Ethernet	VLAN	VLAN
Ethernet	VLAN	VLAN	VLAN	VLAN
VLAN	VLAN	VLAN	VLAN	VLAN

Frame Relay Interworking Restrictions

The following restrictions apply to Frame Relay interworking:

- The attachment circuit maximum transmission unit (MTU) sizes must match when you connect them over MPLS. By default, the MTU size associated with a Frame Relay DLCI is the interface MTU. This may cause problems, for example, when connecting some DLCIs on a Packet over SONET (POS) interface (with a default MTU of 4470 bytes) to Ethernet or VLAN (with a default MTU of 1500 bytes) and other DLCIs on the same PoS interface to ATM (with a default MTU of 4470 bytes). To avoid reducing all the interface MTUs to the lowest common denominator (1500 bytes in this case), you can specify the MTU for individual DLCIs using the **mtu** command.

- Only DLCI mode is supported. Port mode is not supported.
- Configure Frame Relay switching to use DCE or Network-to-Network Interface (NNI). DTE mode does not report status in the Local Management Interface (LMI) process. If a Frame Relay over MPLS circuit goes down and the PE router is in DTE mode, the CE router is never informed of the disabled circuit. You must configure the **frame-relay switching** command in global configuration mode in order to configure DCE or NNI.
- Frame Relay policing is non-distributed on the Cisco 7500 series routers. If you enable Frame Relay policing, traffic is sent to the Route Switch Processor for processing.
- Inverse ARP is not supported with IP interworking. CE routers must use point-to-point subinterfaces or static maps.
- The PE router automatically supports translation of both Cisco and IETF encapsulations that come 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 handle IETF encapsulation on receipt even if it is configured to send Cisco encapsulation.
- With Ethernet interworking, the following translations are supported:
  - Ethernet without LAN FCS (0300800080C20007 or 6558)
  - Spanning tree (0300800080C2000E or 4242)

  All other translations are dropped.
- With IP interworking, IPv4 (03CC or 0800) translation is supported. All other translations are dropped.
- Frame Relay interworking does not pass the IEEE spanning tree bridge protocol data units (BPDUs) when Cisco 7600 series routers are used as PEs.
- PVC status signaling works the same way as in like-to-like case. The PE router reports the PVC status to the CE router based on the availability of the pseudowire. PVC status detected by the PE router will also be reflected into the pseudowire. LMI to OAM interworking is supported when you connect Frame Relay to ATM.

PPP Interworking Restrictions

The following restrictions apply to PPP interworking:

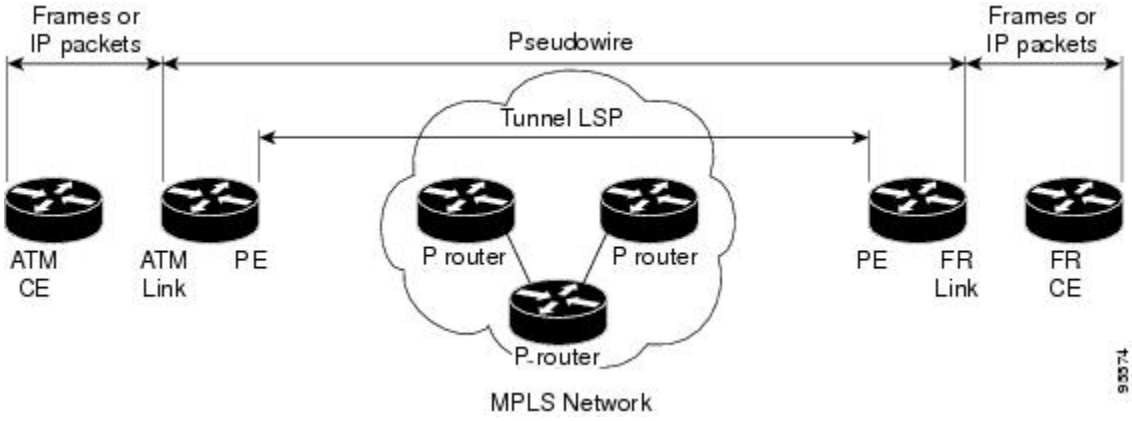
- There must be a one-to-one relationship between a PPP session and the pseudowire. Multiplexing of multiple PPP sessions over the pseudowire is not supported.
- There must be a one-to-one relationship between a PPP session and a Frame Relay DLCI. Each Frame Relay PVC must have only one PPP session.
- Only IP (IPv4 (0021) interworking is supported. Link Control Protocol (LCP) packets and Internet Protocol Control Protocol (IPCP) packets are terminated at the PE router. Everything else is dropped.
- Proxy IPCP is automatically enabled on the PE router when IP interworking is configured on the pseudowire.
- By default, the PE router assumes that the CE router knows the remote CE router's IP address.
- Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP) authentication are supported.

Information About L2VPN Interworking

Overview of L2VPN 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 the different Layer 2 encapsulations. The figure below is an example of Layer 2 interworking, where ATM and Frame Relay packets travel over the MPLS cloud.

Figure 8: Example of ATM to Frame Relay Interworking



The L2VPN Interworking feature supports 802.1Q (VLAN), ATM AAL5, Ethernet, Frame Relay, and PPP attachment circuits over MPLS and L2TPv3. The features and restrictions for like-to-like functionality also apply to L2VPN Interworking.

Note Both AAL5MUX and AAL5SNAP encapsulations are supported. In the case of AAL5MUX, no translation is needed.

L2VPN Interworking Modes

L2VPN Interworking works in either Ethernet (bridged) mode, IP (routed), or Ethernet VLAN mode. You specify the mode by issuing the **interworking** {**ethernet** | **ip** | **vlan**} command in pseudowire class configuration mode.

Ethernet Interworking

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.

Ethernet Interworking is also called bridged interworking. Ethernet frames are bridged across the pseudowire. The CE routers can natively bridge Ethernet or route using a bridged encapsulation model, such as Bridge Virtual Interface (BVI) or RBE. The PE routers operate in Ethernet like-to-like mode. This mode is used to offer the following services:

- LAN services—An example is an enterprise that has several sites, where some sites have Ethernet connectivity to the service provider (SP) network and others have ATM connectivity. The enterprise needs LAN connectivity to all its sites. In this case, traffic from the Ethernet or VLAN of one site can be sent through the IP/MPLS network and encapsulated as bridged traffic over an ATM VC of another site.
- Connectivity services—An example is an enterprise that has different sites that are running an Internal Gateway Protocol (IGP), which has incompatible procedures on broadcast and nonbroadcast links. The enterprise has several sites that are running an IGP, such as Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS), between the sites. In this scenario, some of the procedures (such as a route advertisement or a designated router) depend on the underlying Layer 2 protocol and are different for a point-to-point ATM connection versus a broadcast Ethernet connection. Therefore, the bridged encapsulation over ATM can be used to achieve a homogenous Ethernet connectivity between the CE routers running the IGP.

IP Interworking

The **ip** keyword causes IP packets to be extracted from the attachment circuit and sent over the pseudowire. Attachment circuit frames that do not contain IPv4 packets are dropped.

IP Interworking is also called routed interworking. The CE routers encapsulate IP on the link between the CE and PE routers. A new VC type is used to signal the IP pseudowire in MPLS and L2TPv3. Translation between the Layer 2 and IP encapsulations across the pseudowire is required. Special consideration needs to be given to address resolution and routing protocol operation because these are handled differently on different Layer 2 encapsulations.

IP Interworking is used to provide IP connectivity between sites, regardless of the Layer 2 connectivity to these sites. It is different from a Layer 3 VPN because it is point-to-point in nature, and the service provider does not maintain any customer routing information.

Address resolution is encapsulation-dependent, as explained by the following:

- Ethernet uses ARP.
- Frame Relay and ATM use Inverse ARP.
- PPP uses IPCP.

Therefore, address resolution must be terminated on the PE router. End-to-end address resolution is not supported. Routing protocols operate differently over broadcast and point-to-point media. For Ethernet, the CE routers must either use static routing or configure the routing protocols to treat the Ethernet side as a point-to-point network.

VLAN Interworking

The **vlan** keyword allows the VLAN ID to be included as part of the Ethernet frame. In Cisco IOS Release 12.2(52)SE, you can configure Catalyst 3750 Metro switches to use Ethernet VLAN for Ethernet (bridged) interworking. You can specify the Ethernet VLAN (type 4) by issuing the **interworking vlan** command in pseudowire class configuration mode, which allows the VLAN ID to be included as part of the Ethernet frame.

In releases earlier than Cisco IOS Release 12.2(52)SE, the only way to achieve VLAN encapsulation is to ensure that the CE router is connected to the PE router through an Ethernet VLAN interface or subinterface.

L2VPN Interworking Support Matrix

The supported L2VPN Interworking features are listed in the table below.

Table 15: L2VPN Interworking Supported Features

Feature	MPLS or L2TPv3 Support	IP or Ethernet Support
Ethernet/VLAN to ATM AAL5	MPLS L2TPv3 (12000 series only)	IP Ethernet
Ethernet/VLAN to Frame Relay	MPLS L2TPv3	IP Ethernet
Ethernet/VLAN to PPP	MPLS	IP
Ethernet to VLAN	MPLS L2TPv3	IP Ethernet[8]
Frame Relay to ATM AAL5	MPLS L2TPv3 (12000 series only)	IP
Frame Relay to Ethernet or VLAN	MPLS L2TPv3	IP Ethernet
Frame Relay to PPP	MPLS L2TPv3	IP
L2VPN Interworking: VLAN Enable/Disable Option for AToM	MPLS	Ethernet VLAN

8 With the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature, VLAN interworking can also be supported. For more information, see the "VLAN Interworking" section.

Note On Cisco 12000 series Internet routers:

- Ethernet (bridged) interworking is not supported for L2TPv3.
- IP (routed) interworking is not supported in an L2TPv3 pseudowire that is configured for data sequencing (using the **sequencing** command).

Static IP Addresses for L2VPN Interworking for PPP

For a PE router to perform address resolution with the local CE router for PPP, you can configure the remote CE router's IP address on the PE router. Configure the **ppp ipcp address proxy** command with the remote CE router's IP address on the PE router's xconnect PPP interface. The following is a sample configuration:

```
pseudowire-class ip-interworking
 encapsulation mpls
 interworking ip
interface Serial2/0
```

```
encapsulation ppp
xconnect 10.0.0.2 200 pw-class ip-interworking
ppp ipcp address proxy 10.65.32.14
```

You can also configure the remote CE router's IP address on the local CE router by using the **peer default ip address** command if the local CE router performs address resolution.

How to Configure L2VPN Interworking

Configuring L2VPN Interworking

The L2VPN Interworking feature allows you to connect disparate attachment circuits. Configuring the L2VPN Interworking feature requires that you add the **interworking** command to the list of commands that make up the pseudowire. The steps for configuring the pseudowire for L2VPN Interworking are included in this section. You use the **interworking** command as part of the overall AToM or L2TPv3 configuration. For specific instructions about configuring AToM or L2TPv3, see the following documents:

- *Any Transport over MPLS*
- *Layer 2 Tunneling Protocol Version 3*

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hw-module slot** *slot-number* **np mode feature**
4. **pseudowire-class** *name*
5. **encapsulation** {**mpls** | **l2tpv3**}
6. **interworking** {**ethernet** | **ip** | **vlan**}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hw-module slot** *slot-number* **np mode feature**	(Optional) Enables the L2VPN Interworking feature on the Cisco 12000 series Internet router.

	Command or Action	Purpose
	Example: `Device(config)# hw-module slot 3 np mode feature`	**Note** Enter this command only on a Cisco 12000 series Internet router if you use L2TPv3 for L2VPN Interworking on an ISE (Engine 3) or Engine 5 interface. In this case, you must first enable the L2VPN feature bundle on the line card by entering the **hw-module slot** *slot-number* **np mode feature** command.
Step 4	**pseudowire-class** *name* **Example:** `Device(config)# pseudowire-class class1`	Establishes a pseudowire class with the specified name and enters pseudowire class configuration mode.
Step 5	**encapsulation** {**mpls** | **l2tpv3**} **Example:** `Device(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 6	**interworking** {**ethernet** | **ip** | **vlan**} **Example:** `Device(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it. **Note** On the Cisco 12000 series router, Ethernet (bridged) interworking is not supported for L2TPv3. After you configure the L2TPv3 tunnel encapsulation for the pseudowire using the **encapsulation l2tpv3** command, you cannot enter the **interworking ethernet** command.
Step 7	**end** **Example:** `Device(config-pw)# end`	Exits pseudowire class configuration mode and returns to privileged EXEC mode.

Verifying the L2VPN Interworking Configuration

Use the commands listed in the task below as required. You the commands in the order specified.

SUMMARY STEPS

1. **enable**
2. **show l2tun session all**
3. **show arp**
4. **ping**
5. **show l2tun session interworking**
6. **show mpls l2transport vc detail**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show l2tun session all**
For L2TPv3, you can verify the L2VPN Interworking configuration by using the **show l2tun session all** command on the PE routers.

The following is sample output from the **show l2tun session all** command, and the interworking type is shown in bold.

PE1

```
Device# show l2tun session all

Session Information Total tunnels 1 sessions 1
Session id 15736 is up, tunnel id 35411
Call serial number is 4035100045
Remote tunnel name is PE2
   Internet address is 10.9.9.9
   Session is L2TP signalled
  Session state is established, time since change
 1d22h
   16 Packets sent, 16 received
   1518 Bytes sent, 1230 received
   Receive packets dropped:
     out-of-order:             0
     total:                    0
   Send packets dropped:
     exceeded session MTU:     0
     total:                    0
   Session vcid is 123
  Session Layer 2 circuit, type is Ethernet, name
 is FastEthernet1/1/0
   Circuit state is UP
     Remote session id is 26570, remote tunnel id
 46882
   DF bit off, ToS reflect disabled, ToS value 0,
TTL value 255
   No session cookie information available
   FS cached header information:
     encap size = 24 bytes
       00000000 00000000 00000000 00000000
       00000000 00000000
   Sequencing is off
```

PE2

```
Device# show l2tun session all

Session Information Total tunnels 1 sessions 1
Session id 26570 is up, tunnel id 46882
Call serial number is 4035100045
Remote tunnel name is PE1
   Internet address is 10.8.8.8
   Session is L2TP signalled
   Session state is established, time since
change 1d22h
   16 Packets sent, 16 received
   1230 Bytes sent, 1230 received
   Receive packets dropped:
     out-of-order:             0
     total:                    0
   Send packets dropped:
     exceeded session MTU:     0
     total:                    0
   Session vcid is 123
   Session Layer 2 circuit, type is Ethernet
Vlan, name is FastEthernet2/0.1:10
   Circuit state is UP, interworking type is
Ethernet
    Remote session id is 15736, remote tunnel id
 35411
   DF bit off, ToS reflect disabled, ToS value
0, TTL value 255
   No session cookie information available
   FS cached header information:
     encap size = 24 bytes
       00000000 00000000 00000000 00000000
       00000000 00000000
   Sequencing is off
```

Step 3 **show arp**
You can issue the **show arp** command between the CE routers to ensure that data is being sent.

Example:

```
Device# show arp

Protocol   Address       Age (min)   Hardware Addr    Type    Interface
Internet   10.1.1.5           134    0005.0032.0854   ARPA    FastEthernet0/0
Internet   10.1.1.7             -    0005.0032.0000   ARPA    FastEthernet0/0
```

Step 4 **ping**

You can issue the **ping** command between the CE routers to ensure that data is being sent.

Example:

```
Device# ping 10.1.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Step 5 **show l2tun session interworking**

For L2TPv3, you can verify that the interworking type is correctly set by using the **show l2tun session interworking** command. Enter the command on the PE routers that are performing the interworking translation.

- In the example below, the PE router performs the raw Ethernet translation. The command output displays the interworking type with a dash (-).

Example:

```
Device# show l2tun session interworking

Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    Type IWrk Username, Intf/Vcid, Circuit
15736      35411      10.9.9.9         ETH   -   123,      Fa1/1/0
```

- In the example below, the PE router performs the Ethernet VLAN translation. The command output displays the interworking type as ETH.

Example:

```
Device# show l2tun session interworking

Session Information Total tunnels 1 sessions 1
LocID      TunID      Peer-address    Type IWrk Username, Intf/Vcid, Circuit
26570      46882      10.8.8.8         VLAN ETH  123,      Fa2/0.1:10
```

Step 6 **show mpls l2transport vc detail**

You can verify the AToM configuration by using the **show mpls l2transport vc detail** command. In the following example, the interworking type is shown in bold.

PE1	PE2

Device# **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
	Device# **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

Configuring L2VPN Interworking VLAN Option for AToM

You can specify the Ethernet VLAN (type 4) by issuing the **interworking vlan** command in pseudowire class configuration mode, which allows the VLAN ID to be included as part of the Ethernet frame. In releases earlier than Cisco IOS Release 12.2(52)SE and Cisco IOS Release 12.2(33)SRE, the only way to achieve VLAN encapsulation is to ensure that the CE router is connected to the PE router through an Ethernet link.

Before You Begin

For complete instructions on configuring AToM, see the *Any Transport over MPLS* document.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation** {**mpls** | **l2tpv3**}
5. **interworking** {**ethernet** | **ip** | **vlan**}
6. **end**
7. **show mpls l2transport vc** [**vcid** *vc-id* | **vcid** *vc-id-min vc-id-max*] [**interface** *type number* [*local-circuit-id*]] [**destination** { *ip-address* | *name*}] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *name* **Example:** `Device(config)# pseudowire-class class1`	Establishes a pseudowire class with the specified name and enters pseudowire class configuration mode.
Step 4	**encapsulation** {**mpls**	**l2tpv3**} **Example:** `Device(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 5	**interworking** {**ethernet**	**ip**	**vlan**} **Example:** `Device(config-pw)# interworking vlan`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 6	**end** **Example:** `Device(config-pw)# end`	Exits pseudowire class configuration mode and enters privileged EXEC mode.
Step 7	**show mpls l2transport vc** [**vcid** *vc-id*	**vcid** *vc-id-min vc-id-max*] [**interface** *type number* [*local-circuit-id*]] [**destination** { *ip-address*	*name*}] [**detail**]	Displays information about AToM VCs.

	Command or Action	Purpose
	Example: `Device# show mpls l2transport vc detail`	

Example

When the pseudowire on an interface is different from the VC type, the interworking type is displayed in the **show mpls l2transport vc detail** command output. In the example below, the pseudowire is configured on an Ethernet port and VLAN interworking is configured in the pseudowire class. The relevant output is shown in bold.

```
Device# show mpls l2transport vc 34 detail

Local interface: Et0/1 up, line protocol up, Ethernet up
MPLS VC type is Ethernet, interworking type is Eth VLAN
  Destination address: 10.1.1.2, VC ID: 34, VC status: down
    Output interface: if-?(0), imposed label stack {}
    Preferred path: not configured
    Default path: no route
    No adjacency
  Create time: 00:00:13, last status change time: 00:00:13
  Signaling protocol: LDP, peer unknown
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.2
    Status TLV support (local/remote)   : enabled/None (no remote binding)
      LDP route watch                   : enabled
      Label/status state machine        : local standby, AC-ready, LnuRnd
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: Not sent
      Last local  LDP TLV    status sent: None
      Last remote LDP TLV    status rcvd: None (no remote binding)
      Last remote LDP ADJ    status rcvd: None (no remote binding)
    MPLS VC labels: local 2003, remote unassigned
    Group ID: local 0, remote unknown
    MTU: local 1500, remote unknown
    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
```

Configuration Examples for L2VPN Interworking

Example: Ethernet to VLAN over L2TPv3 (Bridged)

PE1	PE2
ip cef ! l2tp-class interworking-class authentication hostname PE1 password 0 lab ! pseudowire-class inter-ether-vlan encapsulation l2tpv3 interworking ethernet protocol l2tpv3 interworking-class ip local interface Loopback0 ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0 xconnect 10.9.9.9 1 pw-class inter-ether-vlan	ip cef ! l2tp-class interworking-class authentication hostname PE2 password 0 lab ! pseudowire-class inter-ether-vlan encapsulation l2tpv3 interworking ethernet protocol l2tpv3 interworking-class ip local interface Loopback0 ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.3 encapsulation dot1Q 10 xconnect 10.8.8.8 1 pw-class inter-ether-vlan

Example: Ethernet to VLAN over AToM (Bridged)

PE1	PE2
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	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

Example: Frame Relay to VLAN over L2TPv3 (Routed)

PE1	PE2
configure terminal ip cef frame-relay switching ! interface loopback 0 ip address 10.8.8.8 255.255.255.255 no shutdown ! pseudowire-class ip encapsulation l2tpv3 interworking ip ip local interface loopback0 ! interface POS1/0 encapsulation frame-relay clock source internal logging event dlci-status-change no shutdown no fair-queue ! connect fr-vlan POS1/0 206 l2transport xconnect 10.9.9.9 6 pw-class ip ! router ospf 10 network 10.0.0.2 0.0.0.0 area 0 network 10.8.8.8 0.0.0.0 area 0	configure terminal ip routing ip cef frame-relay switching ! interface loopback 0 ip address 10.9.9.9 255.255.255.255 no shutdown ! pseudowire-class ip encapsulation l2tpv3 interworking ip ip local interface loopback0 ! interface FastEthernet1/0/1 speed 10 no shutdown ! interface FastEthernet1/0/1.6 encapsulation dot1Q 6 xconnect 10.8.8.8 6 pw-class ip no shutdown ! router ospf 10 network 10.0.0.2 0.0.0.0 area 0 network 10.9.9.9 0.0.0.0 area 0

Example: Frame Relay to VLAN over AToM (Routed)

PE1	PE2
configure terminal ip cef frame-relay switching ! mpls label protocol ldp mpls ldp router-id loopback0 mpls ip ! pseudowire-class atom encapsulation mpls interworking ip ! interface loopback 0 ip address 10.8.8.8 255.255.255.255 no shutdown ! connect fr-vlan POS1/0 206 l2transport xconnect 10.9.9.9 6 pw-class atom	configure terminal ip routing ip cef frame-relay switching ! mpls label protocol ldp mpls ldp router-id loopback0 mpls ip ! pseudowire-class atom encapsulation mpls interworking ip ! interface loopback 0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface FastEthernet1/0/1.6 encapsulation dot1Q 6 xconnect 10.8.8.8 6 pw-class atom no shutdown

Example: Frame Relay to ATM AAL5 over AToM (Routed)

Note Frame Relay to ATM AAL5 is available only with AToM in IP mode.

PE1	PE2
ip cef frame-relay switching mpls ip mpls label protocol ldp mpls ldp router-id loopback0 force pseudowire-class fratmip encapsulation mpls interworking ip interface Loopback0 ip address 10.33.33.33 255.255.255.255 interface serial 2/0 encapsulation frame-relay ietf frame-relay intf-type dce connect fr-eth serial 2/0 100 l2transport xconnect 10.22.22.22 333 pw-class fratmip interface POS1/0 ip address 10.1.7.3 255.255.255.0 crc 32 clock source internal mpls ip mpls label protocol ldp router ospf 10 passive-interface Loopback0 network 10.33.33.33 0.0.0.0 area 10 network 10.1.7.0 0.0.0.255 area 10	ip cef mpls ip mpls label protocol ldp mpls ldp router-id loopback0 force pseudowire-class fratmip encapsulation mpls interworking ip interface Loopback0 ip address 10.22.22.22 255.255.255.255 interface ATM 2/0 pvc 0/203 l2transport encapsulation aa5snap xconnect 10.33.33.33 333 pw-class fratmip interface POS1/0 ip address 10.1.1.2 255.255.255.0 crc 32 clock source internal mpls ip mpls label protocol ldp router ospf 10 passive-interface Loopback0 network 10.22.22.22 0.0.0.0 area 10 network 10.1.1.0 0.0.0.255 area 10

Example: VLAN to ATM AAL5 over AToM (Bridged)

PE1	PE2
ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface ATM1/0.1 point-to-point pvc 0/100 l2transport encapsulation aal5snap xconnect 10.9.9.9 123 pw-class inter-ether ! interface FastEthernet1/0 xconnect 10.9.9.9 1 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.8.8.8 0.0.0.0 area 0 network 10.1.1.1 0.0.0.0 area 0	ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.1 encapsulation dot1Q 10 xconnect 10.8.8.8 123 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.9.9.9 0.0.0.0 area 0 network 10.1.1.2 0.0.0.0 area 0

Example: Frame Relay to PPP over L2TPv3 (Routed)

PE1	PE2
ip cef ip routing pseudowire-class ppp-fr ! encapsulation l2tpv3 interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.1.1.1 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.1.1 255.255.255.0 ! interface Serial3/0/0 no ip address encapsulation ppp ppp authentication chap ! ip route 10.0.0.0 255.0.0.0 10.16.1.2 ! xconnect 10.2.2.2 1 pw-class ppp-fr ppp ipcp address proxy 10.65.32.14	ip cef ip routing ! frame-relay switching ! pseudowire-class ppp-fr encapsulation l2tpv3 interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.2.2.2 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.2.1 255.255.255.0 1 interface Serial3/0/0 no ip address encapsulation frame-relay frame-relay intf-type dce ! ip route 10.0.0.0 255.0.0.0 10.16.2.2 ! connect ppp-fr Serial3/0/0 100 l2transport xconnect 10.1.1.1 100 pw-class ppp-fr

Example: Frame Relay to PPP over AToM (Routed)

PE1	PE2
ip cef ip routing mpls label protocol ldp mpls ldp router-id loopback0 force ! pseudowire-class ppp-fr encapsulation mpls interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.1.1.1 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.1.1 255.255.255.0 mpls ip label protocol ldp ! interface Serial3/0/0 no ip address encapsulation ppp ppp authentication chap xconnect 10.2.2.2 1 pw-class ppp-fr ppp ipcp address proxy 10.65.32.14 ! ip route 10.0.0.0 255.0.0.0 10.16.1.2	ip cef ip routing mpls label protocol ldp mpls ldp router-id loopback0 force ! frame-relay switching ! pseudowire-class ppp-fr encapsulation mpls interworking ip ip local interface Loopback0 ! interface Loopback0 ip address 10.2.2.2 255.255.255.255 ! interface FastEthernet1/0/0 ip address 10.16.2.1 255.255.255.0 mpls ip mpls label protocol ldp ! interface Serial3/0/0 no ip address encapsulation frame-relay frame-relay intf-type dce ! ip route 10.0.0.0 255.0.0.0 10.16.2.2 ! connect ppp-fr Serial3/0/0 100 l2transport xconnect 10.1.1.1 100 pw-class ppp-fr

Example: Ethernet/VLAN to PPP over AToM (Routed)

PE1	PE2
configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! pseudowire-class ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 no shutdown ! interface POS2/0/1 no ip address encapsulation ppp no peer default ip address ppp ipcp address proxy 10.10.10.1 xconnect 10.9.9.9 300 pw-class ppp-ether no shutdown	configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! encapsulation mpls pseudowire-class ppp-ether interworking ip ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface vlan300 mtu 4470 no ip address xconnect 10.8.8.8 300 pw-class ppp-ether no shutdown ! interface GigabitEthernet6/2 switchport switchport trunk encapsulation dot1q switchport trunk allowed vlan 300 switchport mode trunk no shutdown

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
WAN commands: complete command syntax, command mode, defaults, usage guidelines and examples	Wide-Area Networking Command Reference
Layer 2 Tunnel Protocol Version 3	*Layer 2 Tunneling Protocol Version 3*
Any Transport over MPLS	*Any Transport over MPLS*
Cisco 12000 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.0S*
Cisco 7600 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.2SR*
Cisco 3270 series routers hardware support	*Release Notes for Cisco IOS Software Release 12.2SE*

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-pwe3-frame-relay-03.txt.	*Encapsulation Methods for Transport of Frame Relay over MPLS Networks*
draft-martini-l2circuit-encap-mpls-04.txt.	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-ietf-pwe3-ethernet-encap-08.txt.	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	*Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks*
draft-ietf-ppvpn-l2vpn-00.txt.	*An Architecture for L2VPNs*

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for L2VPN Interworking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

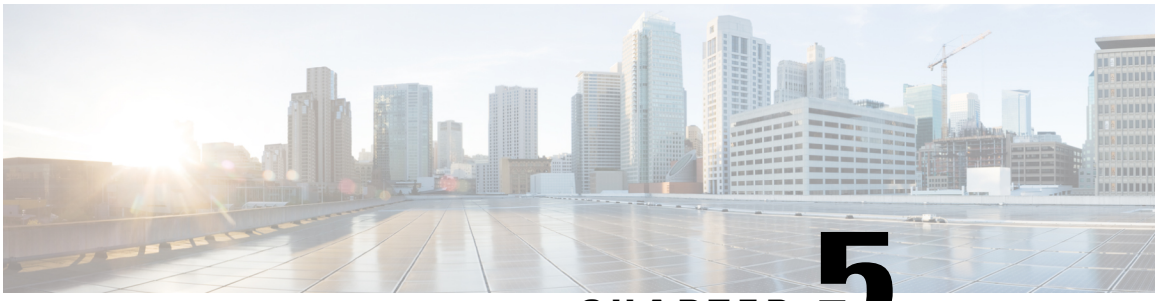
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16: Feature Information for L2VPN Interworking

Feature Name	Releases	Feature Information
L2VPN Interworking	12.0(26)S 12.0(30)S 12.0(32)S 12.0(32)SY 12.2(33)SRA 12.2(33)SRD 12.2(33)SRE 12.2(33)SXH 12.2(52)SE 12.4(11)T	

Feature Name	Releases	Feature Information
		This feature allows disparate attachment circuits to be connected. An interworking function facilitates the translation between the different Layer 2 encapsulations. This feature was introduced in Cisco IOS Release 12.0(26)S. In Cisco IOS Release 12.0(30)S, support was added for Cisco 12000 series Internet routers. In Cisco IOS Release 12.0(32)S, support was added on Engine 5 line cards (SIP-401, SIP-501, SIP-600, and SIP-601) on Cisco 12000 series Internet routers for the following four transport types: • Ethernet/VLAN to Frame Relay Interworking • Ethernet/VLAN to ATM AAL5 Interworking • Ethernet to VLAN Interworking • Frame Relay to ATM AAL5 Interworking On the Cisco 12000 series Internet router, support was added for IP Services Engine (ISE) and Engine 5 line cards that are configured for L2TPv3 tunneling (see the "Layer 2 Tunneling Protocol Version 3" module in *Wide-Area Networking Configuration Guide: Layer 2 Services*). In Cisco IOS Release 12.2(33)SRA, support was added for Cisco 7600 series routers. In Cisco IOS Release 12.4(11)T, support was added for the following transport types: • Ethernet to VLAN Interworking • Ethernet/VLAN to Frame Relay Interworking

Feature Name	Releases	Feature Information
		This feature was integrated into Cisco IOS Release 12.2(33)SXH. In Cisco IOS Release 12.2(33)SRD, support for routed and bridged interworking on SIP-400 was added for Cisco 7600 series routers. In Cisco IOS Release 12.2(52)SE, the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature was added for the Cisco 3750 Metro switch. In Cisco IOS Release 12.2(33)SRE, the L2VPN Interworking: VLAN Enable/Disable Option for AToM feature was added for Cisco 7600 series router. The following commands was introduced or modified: **interworking**.



CHAPTER 5

Layer 2 Local Switching

The Layer 2 Local Switching feature allows you to switch Layer 2 data in two ways:

- Between two interfaces on the same router
- Between two circuits on the same interface port, which is called same-port switching

The interface-to-interface switching combinations supported by this feature are:

- ATM to ATM
- ATM to Ethernet
- ATM to Frame Relay
- Ethernet to Ethernet VLAN
- Frame Relay to Frame Relay (and Multilink Frame Relay in Cisco IOS Release 12.0(28)S and later)
- High-Level Data Link Control (HDLC)

The following same-port switching features are supported:

- ATM Permanent Virtual Circuit (PVC) and Permanent Virtual Path (PVP)
- Ethernet VLAN
- Frame Relay

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Layer 2 Local Switching

- You must enable Cisco Express Forwarding for the Cisco 7200 series router. You must use Cisco Express Forwarding or Distributed Cisco Express Forwarding for the Cisco 7500 series router. (Distributed Cisco Express Forwarding is enabled already by default on the Gigabit Switch Router [GSR]).
- For Frame Relay local switching, you must globally issue the **frame-relay switching** command.

Restrictions for Layer 2 Local Switching

Cisco 7200 and 7500 Series Router Restrictions

- In ATM single cell relay AAL0, the ATM virtual path identifier/virtual channel identifier (VPI/VCI) values must match between the ingress and egress ATM interfaces on the Cisco 7200 series and 7500 series routers. If Layer 2 local switching is desired between two ATM VPIs and VCIs whose values do not match and are on two different interfaces, choose ATM AAL5. However, if the ATM AAL5 is using Operation, Administration, and Maintenance (OAM) transparent mode, the VPI and VCI values must match.
- NSF/SSO: Layer 2 local switching is supported on Cisco 7500 series routers.

Layer 2 local switching is supported on the following interface processors in the Cisco 7200 series routers:

- C7200-I/O-2FE
- C7200-I/O-GE+E (Only the Gigabit Ethernet port of this port adapter is supported.)
- C7200-I/O-FE

Layer 2 local switching is supported on the following interface processors in the Cisco 7500 series routers:

- GEIP (Gigabit Ethernet interface processor)
- GEIP+ (enhanced Gigabit Ethernet interface processor)

Layer 2 local switching is supported on the following port adapters in the Cisco 7200 and 7500 series routers:

- PA-FE-TX (single-port Fast Ethernet 100BASE-TX)

- PA-FE-FX (single-port Fast Ethernet 100BASE-FX)
- PA-2FE-TX (dual-port Fast Ethernet 100BASE-TX)
- PA-2FE-FX (dual-port Fast Ethernet 100BASE-FX)
- PA-4E (4-port Ethernet adapter)
- PA-8E (8-port Ethernet adapter)
- PA-4T (4-port synchronous serial port adapter)
- PA-4T+ (enhanced 4-port synchronous serial port adapter)
- PA-8T (8-port synchronous serial port adapter)
- PA-12E/2FE (12-port Ethernet/2-port Fast Ethernet (FE) adapter) [Cisco 7200 only]
- PA-GE (Gigabit Ethernet port adapter) [Cisco 7200 only]
- PA-H (single-port High-Speed Serial Interface (HSSI) adapter)
- PA-2H (dual-port HSSI adapter)
- PA-MC-8E1 (8-port multichannel E1 G.703/G.704 120-ohm interfaces)
- PA-MC-2EI (2-port multichannel E1 G.703/G.704 120-ohm interfaces)
- PA-MC-8T1 (8-port multichannel T1 with integrated data service units (DSUs) and channel service units CSUs))
- PA-MC-4T1 (4-port multichannel T1 with integrated CSUs and DSUs)
- PA-MC-2T1 (2-port multichannel T1 with integrated CSUs and DSUs)
- PA-MC-8TE1+ (8-port multichannel T1/E1)
- PA-MC-T3 (1-port multichannel T3 interface)
- PA-MC-E3 (1-port multichannel E3 interface)
- PA-MC-2T3+ (2-port enhanced multichannel T3 port adapter)
- PA-MC-STM1 (1-port multichannel STM-1 port adapter) [Cisco 7500 only]
- PA-T3 (single-port T3 port adapter)
- PA-E3 (single-port E3 port adapter)
- PA-2E3 (2-port E3 port adapter)
- PA-2T3 (2-port T3 port adapter)
- PA-POS-OC-3SML (single-port Packet over SONET (POS), single-mode, long reach)
- PA-POS-OC-3SMI (single-port PoS, single-mode, intermediate reach)
- PA-POS-OC-3MM (single-port PoS, multimode)
- PA-A3-OC-3 (1-port ATM OC-3/STM1 port adapter, enhanced)
- PA-A3-OC-12 (1-port ATM OC-12/STM-4 port adapter, enhanced) [Cisco 7500 only]
- PA-A3-T3 (DS3 high-speed interface)
- PA-A3-E3 (E3 medium-speed interface)

- PA-A3-8T1IMA (ATM inverse multiplexer over ATM port adapter with 8 T1 ports)
- PA-A3-8E1IMA (ATM inverse multiplexer over ATM port adapter with 8 E1 ports)
- PA-A6 (Cisco ATM Port Adapter)

Cisco 7600 and 6500 Series Router Restrictions

- Layer 2 local switching supports the following port adapters and interface processors on the Cisco 7600-SUP720/MSFC3 router:
  - All port adapters on the Enhanced FlexWAN module
  - All shared prot adaptors (SPAs) on the SIP-200 line cards
- On the Cisco 6500 series and 7600 series routers, only *like-to-like* local switching is supported (ATM to ATM and Frame Relay to Frame Relay).
- Same-port switching is not supported on the Cisco 6500 series and 7600 series routers.

Cisco 10000 Series Router Restrictions

For information about Layer 2 local switching on the Cisco 10000 series routers, see the *Configuring Layer 2 Local Switching* document.

Gigabit Switch Router Restrictions

- VPI/VCI rewrite is supported.
- All GSR line cards support Frame Relay-to-Frame Relay local switching.
- 8-port OC-3 ATM Engine 2 line cards support only like-to-like Layer 2 local switching.
- IP Service Engine (ISE) (Engine 3) line cards support like-to-like and any-to-any local switching. Non-ISE line cards support only like-to-like local switching.

Starting in Cisco IOS Release 12.0(31)S2, ISE customer edge-facing interfaces support the following types of like-to-like and any-to-any local switching:

-
  - ATM to ATM
  - ATM to Ethernet
  - ATM to Frame Relay
  - Ethernet to Ethernet VLAN
  - Frame Relay to Frame Relay (including Multilink Frame Relay)
  - Same-port switching for ATM (PVC and PVP)
  - Same-port switching for Ethernet VLAN
  - Same-port switching for Frame Relay

Note Native Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunnel sessions on customer edge-facing line cards can coexist with tunnel sessions that use a tunnel-server card.

- Starting in Cisco IOS Release 12.0(32)SY, customer edge-facing interfaces on Engine 5 SPAs and SPA Interface Processors (SIPs) support the following types of like-to-like local switching:
  - Ethernet to Ethernet VLAN
  - Frame Relay to Frame Relay (including Multilink Frame Relay)
  - Same-port switching for Ethernet VLAN
  - Same-port switching for Frame Relay
- For ATM-to-ATM local switching, the following ATM types are supported for the Layer 2 Local Switching feature:
  - ATM adaptation layer 5 (AAL5)
  - ATM single cell relay adaptation layer 0 (AAL0), VC mode
  - ATM single cell relay VP mode on the GSR
  - ATM single cell relay VC and VP modes on ISE line cards on the GSR
- Starting with Cisco IOS Release 12.0(30)S, you can use local switching and cell packing with ATM VP or VC mode on the GSR on IP Services Engine (ISE/Engine 3) line cards. For information about how to configure cell packing, refer to Any Transport over MPLS.

Unsupported Hardware

The following hardware is not supported:

- Cisco 7200—non-VXR chassis
- Cisco 7500—Route Switch Processor (RSP)1 and 2
- Cisco 7500—Versatile Interface Processor (VIP) 2-40 and below
- GSR—4-port OC-3 ATM Engine-0 line card
- GSR—4-port OC-12 ATM Engine-2 line card
- GSR—1-port OC-12 ATM Engine-0 line card
- GSR—Ethernet Engine-1, Engine-2, and Engine-4 line cards

Information About Layer 2 Local Switching

Layer 2 Local Switching Overview

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) or between interfaces of different types (for example, Frame Relay to ATM) on the same router. The interfaces can be on the same line card or on two different cards. During these kinds of switching, the Layer 2 address is used, not any Layer 3 address.

Additionally, same-port local switching allows you to switch Layer 2 data between two circuits on the same interface.

NSF SSO—Local Switching Overview

Nonstop forwarding (NSF) and stateful switchover (SSO) improve the availability of the network by providing redundant Route Processors (RPs) and checkpointing of data to ensure minimal packet loss when the primary RP goes down. NSF/SSO support is available for the following locally switched attachment circuits:

- Ethernet to Ethernet VLAN
- Frame Relay to Frame Relay

Layer 2 Local Switching Applications

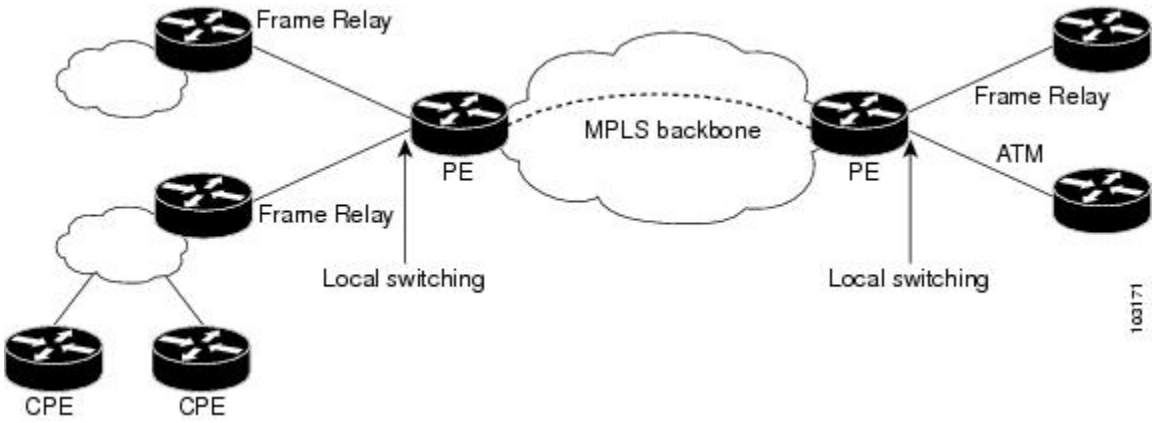
Incumbent local exchange carriers (ILECs) who use an interexchange carrier (IXC) to carry traffic between two local exchange carriers can use the Layer 2 Local Switching feature. Telecom regulations require the ILECs to pay the IXCs to carry that traffic. At times, the ILECs cannot terminate customer connections that are in different local access and transport areas (LATAs). In other cases, customer connections terminate in the same LATA, which may also be on the same router.

For example, company A has more than 50 LATAs across the country and uses three routers for each LATA. Company A uses companies B and C to carry traffic between local exchange carriers. Local switching of Layer 2 frames on the same router might be required.

Similarly, if a router is using, for example, a channelized interface, it might need to switch incoming and outgoing traffic across two logical interfaces that reside on a single physical port. The same-port local switching feature addresses that implementation.

The figure below shows a network that uses local switching for both Frame Relay to Frame Relay and ATM to Frame Relay local switching.

Figure 9: Local Switching Example



Access Circuit Redundancy Local Switching

The Automatic Protection Switching (APS) mechanism provides a switchover time of less than 50 milliseconds. However, the switchover time is longer in a pseudowire configuration due to the time the pseudowire takes to enter the UP state on switchover. The switchover time of the pseudowire can be eliminated if there is a single pseudowire on the working and protect interfaces instead of separate pseudowire configurations. A single pseudowire also eliminates the need to have Label Distribution Protocols (LDP) negotiations on a switchover. The virtual interface or controller model provides a method to configure a single pseudowire between the provider edge (PE) routers.

Access Circuit Redundancy (ACR) ensures low data traffic downtime by reducing the switchover time. ACR works on the APS 1+1, nonrevertive model where each redundant line pair consists of a working line and a protect line. If a signal fail condition or a signal degrade condition is detected, the hardware switches from the working line to the protect line.

The working and protect interfaces can be on the following:

- Same SPA
- Different SPA but on the same line card
- SPAs on different line cards

When the working or protection interface is configured with ACR, a virtual interface is created and a connection is established between the virtual interfaces to facilitate the switching of data between the interfaces.

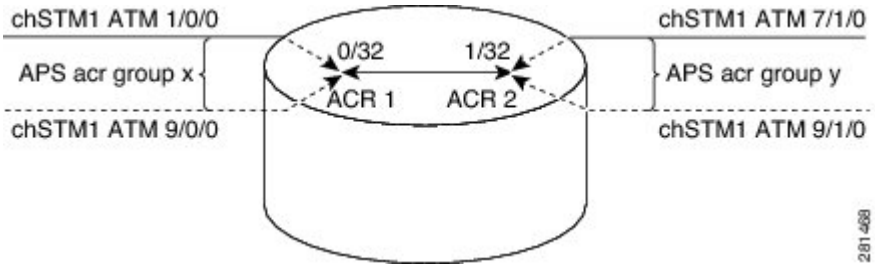
ACR for ATM-to-ATM Local Switching

ACR for ATM-to-ATM local switching supports the ATM AAL5 and ATM AAL0 encapsulation types and switches Layer 2 data between L2 transport virtual circuits (VCs).

Note The L2 transport VCs must be configured with the same encapsulation type.

The figure below shows the ACR for ATM-to-ATM local switching model.

Figure 10: ATM-to-ATM ACR Local Switching Model



In the figure:

- ATM 1/0/0 and ATM 9/0/0 are configured as working and protection interfaces of ACR 1 group.
- ATM 7/1/0 and ATM 9/1/0 are configured as working and protection interfaces of ACR 2 group.
- A connection is established between the ACRs.
- The Add/Drop Multiplexer (ADM) sends data to both the interfaces, which are part of the ACR group ACR 1.
- The cells or packets received on the APS active interface VC (0/32) of ACR group 1 are switched to the ACR 2 interface VC (1/32) and the cells or packets from the APS inactive interface are dropped.
- The packets received on the ACR 2 VC (1/32) interface are replicated on both the physical interfaces, which are part of the ACR group ACR 2.

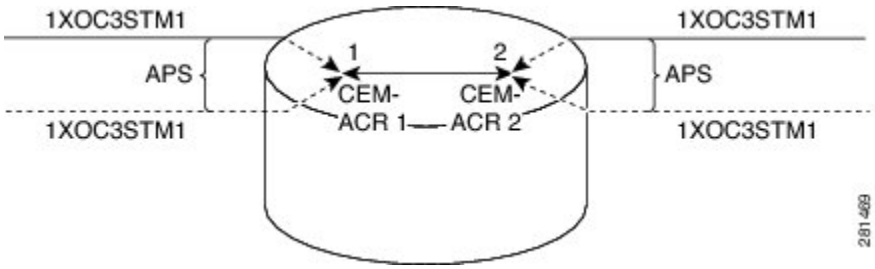
ACR for CEM-to-CEM Local Switching

Circuit Emulation (CEM) transports Time Division Multiplexing (TDM) data over TDM pseudowires, allowing mobile operators to carry TDM traffic over an IP or Multiprotocol Label Switching (MPLS) network. ACR for CEM-to-CEM involves creating a virtual controller and associating the virtual controller with the physical controllers. The virtual controller is created when APS and ACR are configured on the physical controller. All commands executed on the virtual controller apply to the working and protect controller. The virtual controller simplifies the single point of configuration and provides the flexibility of not running a backup pseudowire for the protect controller in the event of a failure. This way there is no switchover between the pseudowires, which in turn reduces the recovery time when the physical link fails.

When the CEM group is configured on the virtual controller, a virtual CEM-ACR interface is created and associated with the CEM circuit. ACR creates CEM interfaces and CEM circuits on the physical interfaces that correspond to the physical controllers belonging to the same ACR group.

The figure below shows the ACR for CEM-to-CEM local switching model:

Figure 11: CEM-to-CEM ACR Local Switching Model



In the figure:

- Packets are received from the ADM. The packets from the APS inactive interface are dropped and the packets received on the APS active interface are switched.
- The packets received on the CEM circuit ID 1 of the APS active interface, which is part of ACR group 1, are switched to the CEM circuit ID 2 of the APS active interface, which is part of ACR group 2.
- The packets are duplicated and sent on both the APS active and inactive physical CEM interfaces that are part of ACR group 2.

How to Configure Layer 2 Local Switching

For information about Layer 2 local switching on the Cisco 10000 series routers, see the *Configuring Layer 2 Local Switching* document.

Configuring ATM-to-ATM PVC Local Switching and Same-Port Switching

You can configure local switching for both ATM AAL5 and ATM AAL0 encapsulation types.

Creating the ATM PVC is not required. If you do not create a PVC, one is created for you. For ATM-to-ATM local switching, the autoprovisioned PVC is given the default encapsulation type AAL0 cell relay.

Note Starting with Cisco IOS Release 12.0(30)S, you can configure same-port switching following the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **pvc** *vpi* **/** *vci* **l2transport**
5. **encapsulation** *layer-type*
6. **exit**
7. **exit**
8. **connect** *connection-name interface pvc interface pvc*

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 atm** *slot/port* **Example:** `Router(config)# interface atm1/0/0`	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
Step 4	**pvc** *vpi* **/** *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/100 l2transport`	Assigns a VPI and VCI and enters ATM PVC l2transport configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation** *layer-type* **Example:** `Router(cfg-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies the encapsulation type for the ATM PVC. Both AAL0 and AAL5 are supported. • Repeat Steps 3 through 5 for another ATM PVC on the same router.
Step 6	**exit** **Example:** `Router(cfg-if-atm-l2trans-pvc)# exit`	Exits PVC l2transport configuration mode and returns to interface configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**connect** *connection-name interface pvc interface pvc* **Example:** `Router(config)# connect atm-con atm1/0/0 1/100 atm2/0/0 1/100`	Creates a local connection between the two specified permanent virtual circuits.

Configuring ATM-to-ATM PVP Local Switching

Perform this task to configure ATM-to-ATM PVP local switching.

Starting with Cisco IOS Release 12.0(30)S, you can configure same-port switching, as detailed in the Configuring ATM PVP Same-Port Switching, on page 188.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **atm pvp** *vpi* **l2transport**
5. **exit**
6. **exit**
7. **connect** *connection-name interface pvp interface pvp*

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	**interface atm** *slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 100 l2transport`	Identifies the virtual path and enters PVP l2transport configuration mode. The **l2transport**keyword indicates that the PVP is a switched PVP instead of a terminated PVP. • Repeat Steps 3 and 4 for another ATM permanent virtual path on the same router.
Step 5	**exit** **Example:** `Router(config-if-atm-l2trans-pvp)# exit`	Exits PVP l2transport configuration mode and returns to interface configuration mode.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**connect** *connection-name interface pvp interface pvp* **Example:** `Router(config)# connect atm-con atm1/0 100 atm2/0 200`	Creates a local connection between the two specified permanent virtual paths.

Configuring ATM PVP Same-Port Switching

Perform this task to configure ATM PVP switching on an ATM interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port*
4. **atm pvp** *vpi* **l2transport**
5. **exit**
6. **exit**
7. **connect** *connection-name interface pvp interface pvp*

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 atm** *slot* / *subslot* / *port* **Example:** `Router(config)# interface atm1/0/0`	Specifies an ATM line card, subslot (if available), and port, and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 100 l2transport`	Specifies one VPI and enters PVP l2transport configuration mode. Repeat this step for the other ATM permanent virtual path on this same port. • The **l2transport**keyword indicates that the indicated PVP is a switched PVP instead of a terminated PVP.
Step 5	**exit** **Example:** `Router(config-if-atm-l2trans-pvp)# exit`	Exits PVP l2transport configuration mode and returns to interface configuration mode.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**connect** *connection-name interface pvp interface pvp* **Example:** `Router(config)# connect atm-con atm1/0/0 100 atm1/0/0 200`	In global configuration mode, creates the local connection between the two specified permanent virtual paths.

Configuring ATM-to-Ethernet Port Mode Local Switching

For ATM to Ethernet port mode local switching, creating the ATM PVC is not required. If you do not create a PVC, one is created for you. For ATM-to-Ethernet local switching, the autoprovisioned PVC is given the default encapsulation type AAL5SNAP.

ATM-to-Ethernet local switching supports both the IP and Ethernet interworking types. When the Ethernet interworking type is used, the interworking device (router) expects a bridged packet. Therefore, configure the ATM CPE for either IRB or RBE.

Note Enabling ICMP Router Discovery Protocol on the Ethernet side is recommended.

ATM-to-Ethernet local switching supports the following encapsulation types:

- ATM-to-Ethernet with IP interworking: AAL5SNAP, AAL5MUX
- ATM-to-Ethernet with Ethernet interworking: AAL5SNAP

Perform this task to configure local switching between ATM and Ethernet port mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/port*
4. **pvc** *vpi* / *vci* **l2transport**
5. **encapsulation** *layer-type*
6. **exit**
7. **exit**
8. **interface fastethernet** *slot* / *subslot* / *port*
9. **exit**
10. **connect** *connection-name interface pvc interface* [**interworkingip** | **ethernet**]

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	**interface atm** *slot/port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
Step 4	**pvc** *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Assigns a VPI and VCI and enters PVC l2transport configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation** *layer-type* **Example:** `Router(cfg-if-atm-l2trans-pvc)# encapsulation aal5snap`	Specifies the encapsulation type for the PVC.
Step 6	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits PVC l2transport configuration mode and returns to interface configuration mode.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface fastethernet** *slot* / *subslot* / *port* **Example:** `Router(config)# interface fastethernet6/0/0`	Specifies a Fast Ethernet line card, subslot (if available), and port, and enters interface configuration mode.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**connect** *connection-name interface pvc interface* [**interworkingip** | **ethernet**] **Example:** `Router(config)# connect atm-eth-con atm1/0 0/100 fastethernet6/0/0 interworking ip`	Creates a local connection between the two interfaces and specifies the interworking type. • Both the IP and Ethernet interworking types are supported.

Configuring ATM-to-Ethernet VLAN Mode Local Switching

For ATM-to-Ethernet VLAN mode local switching, creating the ATM permanent virtual circuit (PVC) is not required. If you do not create a PVC, one is created for you. For ATM-to-Ethernet local switching, the autoprovisioned PVC is given the default encapsulation type as ATM adaptation layer 5 (AAL5) Subnetwork Access Protocol (SNAP).

ATM-to-Ethernet local switching supports both the IP and Ethernet interworking types. When the Ethernet interworking type is used, the interworking device (router) expects a bridged packet. Therefore, configure the ATM customer premises equipment (CPE) for either Integrated Routing and Bridging (IRB) or Routed Bridged Encapsulation (RBE).

ATM-to-Ethernet local switching supports the following encapsulation types:

- ATM-to-Ethernet with IP interworking: AAL5SNAP, AAL5 multiplexer (MUX)
- ATM-to-Ethernet with Ethernet interworking: AAL5SNAP

Note Enabling Internet Control Message Protocol (ICMP) Router Discovery Protocol (IRDP) on the Ethernet side is recommended.

The VLAN header is removed from frames that are received on an Ethernet subinterface.

Note On the provider edge (PE) router, ensure that the maximum transmission unit (MTU) value of ATM interfaces (default MTU is 4470 bytes) and Gigabit Ethernet interfaces (default MTU is 1500 bytes) is the same. On the customer edge (CE) router, ensure that the MTU value of ATM and Gigabit Ethernet interfaces is at least 14 bytes less than the MTU value of the respective interfaces on the PE router during ATM-to-Ethernet VLAN mode local switching.

Perform this task to configure local switching for ATM to Ethernet in VLAN mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot*/*subslot*/*port*
4. **pvc** *vpi*/*vci* **l2transport**
5. **encapsulation** *layer-type*
6. **exit**
7. **interface fastethernet** *slot*/*port*.*subinterface-number*
8. **encapsulation dot1q** *vlan-id*
9. **exit**
10. **connect** *connection-name interface pvc interface* [**interworking ip** | **ethernet**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot*/*subslot*/*port* **Example:** `Device(config)# interface atm1/0/0`	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **l2transport** **Example:** `Device(config-if)# pvc 1/200 l2transport`	Assigns a VPI and VCI and enters PVC l2transport configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation** *layer-type* **Example:** `Device(cfg-if-atm-l2trans-pvc)# encapsulation aal5snap`	Specifies the encapsulation type for the PVC.
Step 6	**exit** **Example:** `Device(cfg-if-atm-l2trans-pvc)# exit`	Exits PVC l2transport configuration mode and returns to interface configuration mode.
Step 7	**interface fastethernet** *slot*/*port*.*subinterface-number* **Example:** `Device(config-if)# interface fastethernet6/0/0.1`	Specifies a Fast Ethernet line card, subslot (if available), port and subinterface and enters subinterface configuration mode.
Step 8	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100`	Enables the interface to accept 802.1Q VLAN packets.
Step 9	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**connect** *connection-name interface pvc interface* [**interworking ip** | **ethernet**] **Example:** `Device(config)# connect atm-eth-vlan-con atm1/0/0 0/100 fastethernet6/0/0.1 interworking ip`	In global configuration mode, creates a local connection between the two interfaces and specifies the interworking type. • Both the IP and Ethernet interworking types are supported.

Configuring Ethernet VLAN Same-Port Switching

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot*/*port*.*subinterface-number*
4. **encapsulation dot1q** *vlan-id*
5. **exit**
6. **interface fastethernet** *slot* / *port.subinterface-number*
7. **encapsulation dot1q** *vlan-id*
8. **exit**
9. **connect** *connection-name interface interface*

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 fastethernet** *slot*/*port*.*subinterface-number* **Example:** `Router(config)# interface fastethernet6/0.1`	Specifies the first Fast Ethernet line card, subslot (if available), port, and subinterface and enters subinterface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 10`	Enables that subinterface to accept 802.1Q VLAN packets and specifies the first VLAN.
Step 5	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 6	**interface fastethernet** *slot* / *port.subinterface-number* **Example:** `Router(config)# interface fastethernet6/0.2`	Specifies the second Fast Ethernet line card, subslot (if available), port, and subinterface and enters subinterface configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 20`	Enables this subinterface to accept 802.1Q VLAN packets and specifies the second VLAN.
Step 8	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 9	**connect** *connection-name interface interface* **Example:** `Router(config)# connect conn fastethernet6/0.1 fastethernet6/0.2`	Creates a local connection between the two subinterfaces (and hence their previously specified VLANs) on the same Fast Ethernet port.

Configuring Ethernet Port Mode to Ethernet VLAN Local Switching

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot*/*subslot*/*port*
4. **interface fastethernet** *slot*/*port*/*subinterface-number*
5. **encapsulation dot1q** *vlan-id*
6. **exit**
7. **connect** *connection-name interface interface* [**interworking ip** | **ethernet**]

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 fastethernet** *slot/subslot/port* **Example:** `Router(config)# interface fastethernet3/0/0`	Specifies a Fast Ethernet line card, subslot (if available), and port and enters interface configuration mode. This is the interface on one side of the PE router that passes Ethernet packets to and from the customer edge (CE) router.
Step 4	**interface fastethernet** *slot/port/subinterface-number* **Example:** `Router(config)# interface fastethernet6/0/0.1`	Specifies a Fast Ethernet line card, subslot (if available), port, and subinterface and enters subinterface configuration mode. This is the interface on the other side of the PE router than passes Ethernet VLAN packets to and from the CE router.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the interface to accept 802.1Q VLAN packets.
Step 6	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 7	**connect** *connection-name interface interface* [**interworking ip** | **ethernet**] **Example:** `Router(config)# connect eth-ethvlan-con fastethernet3/0/0 fastethernet6/0/0.1 interworking ip`	Creates a local connection between the two interfaces and specifies the interworking type. • Both the IP and Ethernet interworking types are supported.

Configuring ATM-to-Frame Relay Local Switching

You use the **interworking ip** keywords for configuring ATM-to-Frame Relay local switching.

FRF.8 Frame Relay-to-ATM service interworking functionality is not supported. Frame Relay discard-eligible (DE) bits do not get mapped to ATM cell loss priority (CLP) bits, and forward explicit congestion notification (FECN) bits do not get mapped to ATM explicit forward congestion indication (EFCI) bits.

Creating the PVC is not required. If you do not create a PVC, one is created for you. For ATM-to-Ethernet local switching, the automatically provisioned PVC is given the default encapsulation type AAL5SNAP.

ATM-to-Frame Relay local switching supports the following encapsulation types:

- AAL5SNAP
- AAL5NLPID (GSR uses AAL5MUX instead, for IP interworking)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot*/*port*
4. **pvc** *vpi*/*vci* **l2transport**
5. **encapsulation** *layer-type*
6. **exit**
7. **interface serial** *slot*/*subslot*/*port*
8. **encapsulation frame-relay** [**cisco** | **ietf**]
9. **frame-relay interface-dlci** *dlci* **switched**
10. **exit**
11. **connect** *connection-name interface pvc interface dlci* [**interworking ip** | **ethernet**]

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 atm** *slot*/*port* **Example:** `Router(config)# interface atm1/0`	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Assigns a VPI and VCI and enters PVC l2transport configuration mode.

	Command or Action	Purpose
		• The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation** *layer-type* **Example:** `Router(cfg-if-atm-l2trans-pvc)# encapsulation aal5snap`	Specifies the encapsulation type for the PVC.
Step 6	**exit** **Example:** `Router(cfg-if-atm-l2trans-pvc)# exit`	Exits PVC l2transport configuration mode and returns to interface configuration mode.
Step 7	**interface serial** *slot/subslot/port* **Example:** `Router(config-if)# interface serial6/0/0`	Specifies a channelized line card, subslot (if available), and serial port.
Step 8	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. • The encapsulation type does not matter for local switching. It has relevance only for terminated circuits.
Step 9	**frame-relay interface-dlci** *dlci* **switched** **Example:** `Router(config-if)# frame-relay interface-dlci 100 switched`	(Optional) Configures a switched Frame Relay DLCI. • If you do not create a Frame Relay PVC in this step, one is automatically created by the **connect** command.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**connect** *connection-name interface pvc interface dlci* [**interworking ip** | **ethernet**] **Example:** `Router(config)# connect atm-fr-con atm1/0 0/100 serial6/0/0 100 interworking ip`	Creates a local connection between the two interfaces.

Configuring Frame Relay-to-Frame Relay Local Switching

For information on Frame Relay-to-Frame Relay Local Switching, see the Distributed Frame Relay Switching feature module.

With Cisco IOS Release 12.0(30)S, you can switch between virtual circuits on the same port, as detailed in the Configuring Frame Relay Same-Port Switching, on page 200.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **frame-relay switching**
5. **interface** *type number*
6. **encapsulation frame-relay** [**cisco** | **ietf**]
7. **frame-relay interface-dlci** *dlci* **switched**
8. **exit**
9. **exit**
10. **connect** *connection-name interface dlci interface dlci*

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 cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding operation. • For the Cisco 7500 series router, use the **ip cef distributed** command. (On the GSR, this command is already enabled by default). • For the Cisco 7200 series router, use the **ip cef** command.
Step 4	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay DCE device or a Network-to-Network Interface (NNI).
Step 5	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Specifies a Frame Relay interface and enters interface configuration mode.
Step 6	**encapsulation frame-relay** [**cisco**	**ietf**]	Enables Frame Relay encapsulation.

	Command or Action	Purpose
	Example: `Router(config-if)# encapsulation frame-relay`	• The default is **cisco** encapsulation. • You do not need to specify an encapsulation type.
Step 7	**frame-relay interface-dlci** *dlci* **switched** **Example:** `Router(config-if)# frame-relay interface-dlci 100 switched`	(Optional) Creates a switched PVC and enters Frame Relay DLCI configuration mode. • Repeat Steps 5 through 7 for each switched PVC. • If you do not create a Frame Relay PVC in this step, it will automatically be created by the **connect** command.
Step 8	**exit** **Example:** `Router(config-fr-dlci)# exit`	Exits Frame Relay DLCI configuration mode and returns to interface configuration mode.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**connect** *connection-name interface dlci interface dlci* **Example:** `Router(config)# connect connection1 serial0 100 serial1 101`	Defines a connection between Frame Relay PVCs.

Configuring Frame Relay Same-Port Switching

Perform this task to configure Frame Relay switching on the same interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **frame-relay switching**
5. **interface** *type number*
6. **encapsulation frame-relay** [**cisco** | **ietf**]
7. **frame-relay intf-type** [**dce**| **dte**| **nni**]
8. **frame-relay interface-dlci** *dlci* **switched**
9. **exit**
10. **exit**
11. **connect** *connection-name interface dlci interface dlci*

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 cef** [**distributed**] **Example:** `Router(config)# ip cef`	Enables Cisco Express Forwarding operation. • For the Cisco 7500 series router, use the **ip cef distributed** command. (On the GSR, this command is already enabled by default). • For the Cisco 7200 series router, use the **ip cef** command.
Step 4	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay DCE device or a NNI.
Step 5	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Specifies a Frame Relay interface and enters interface configuration mode.
Step 6	**encapsulation frame-relay** [**cisco**	**ietf**]	Enables Frame Relay encapsulation.

	Command or Action	Purpose
	Example: `Router(config-if)# encapsulation frame-relay`	• The default is **cisco** encapsulation. • You do not need to specify an encapsulation type.
Step 7	**frame-relay intf-type** [**dce**| **dte**| **nni**] **Example:** `Router(config-if)# frame-relay intf-type nni`	(Optional) Enables support for a particular type of connection: • DCE • DTE (default) • NNI
Step 8	**frame-relay interface-dlci** *dlci* **switched** **Example:** `Router(config-if)# frame-relay interface-dlci 100 switched`	(Optional) Creates a switched PVC and enters Frame Relay DLCI configuration mode. • If you do not create a Frame Relay PVC in this step, it will automatically be created by the **connect** command.
Step 9	**exit** **Example:** `Router(config-fr-dlci)# exit`	Exits Frame Relay DLCI configuration mode and returns to interface configuration mode.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**connect** *connection-name interface dlci interface dlci* **Example:** `Router(config)# connect connection1 serial1/0 100 serial1/0 200`	Defines a connection between the two data links.

Configuring HDLC Local Switching

Perform this task to configure local switching for HDLC. The PE routers are configured with HDLC encapsulation. The CE routers are configured with any HDLC-based encapsulation, including HDLC, PPP, and Frame Relay.

Before You Begin

- Ensure that the interfaces you configure for HDLC encapsulation can handle ping packets that are smaller, the same size as, or larger than the CE interface MTU.
- Enable Cisco Express Forwarding.

Note Do not configure other settings on the interfaces configured for HDLC encapsulation. If you assign an IP address on the interface, the **connect** command is rejected and the following error message displays:
`Incompatible with IP address command on interface - command rejected.`
If you configure other settings on the interface that is enabled for HDLC encapsulation, the local switching feature may not work.

- Interworking is not supported.
- Same-port local switching for HDLC is not supported.
- Dialer and ISDN interfaces are not supported. Only serial, HSSI, and POS interfaces can be configured for HDLC local switching.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **interface** *type number*
5. **exit**
6. **connect** *connection-name interface interface*

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 cef** **Example:** `Router(config)# ip cef`	Enables Cisco Express Forwarding operation.
Step 4	**interface** *type number* **Example:** `Router(config)# interface serial 2/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**connect** *connection-name interface interface* **Example:** `Router(config)# connect connection1 serial1/0 serial1/0`	Defines a connection between the two interfaces.

Configuring ACR for ATM-to-ATM Local Switching

Note The **connect** command provides an infrastructure to create the required L2 transport VCs with the default AAl0 encapsulation type and does not require that the VCs must exist.

Perform this task to configure ACR for ATM-to-ATM local switching.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot*/*subslot*/*port*
4. **aps group** [**acr**] *group-number*
5. **aps working** *circuit-number*
6. **aps protect** *circuit-number ip-address*
7. **exit**
8. **interface acr** *acr-group-number*
9. **pvc** [*name*] *vpi*/*vci* **l2transport**
10. **exit**
11. **exit**
12. **connect** *connection-name type number pvc type number pvc*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot/subslot/port* **Example:** `Router(config)# interface atm8/0/0`	Specifies an ATM line card, a subslot (if available), and a port and enters interface configuration mode.
Step 4	**aps group** [**acr**] *group-number* **Example:** `Router(config-if)# aps group acr 1`	Configures an ACR working and protect interface. • *group-number*—Number of the group.
Step 5	**aps working** *circuit-number* **Example:** `Router(config-if)# aps working 1`	Enables an ATM OC-3 interface as the working interface. • *circuit-number*—Number of the circuit that will be enabled as the working interface. Repeat Steps 3 to 5 for the protect interface.
Step 6	**aps protect** *circuit-number ip-address* **Example:** `Router(config-if)# aps protect 1 10.0.0.1`	Enables an ATM OC-3 interface as the protect interface. • *circuit-number*—Number of the circuit that will be enabled as the protect interface. • *ip-address*—IP address of the router that has the working ATM OC-3 interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface acr** *acr-group-number* **Example:** `Router(config)# interface acr 1`	Specifies an ACR interface and enters interface configuration mode. • *acr-group-number*—The group number assigned to the working and protect interface.
Step 9	**pvc** [*name*] *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 0/32 l2transport`	Creates an ATM PVC and enters ATM virtual circuit configuration mode.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-if-atm-vc)# exit`	Exits ATM virtual circuit configuration mode and returns to interface configuration mode. Repeat Steps 8 and 9 for the other ACR group.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**connect** *connection-name type number pvc type number pvc* **Example:** `Router(config)# connect connection1 acr 1 0/32 acr 2 1/32`	Defines the connection between the ATM-ACR interfaces. • *connection-name*—Local switching connection name. • *type*—Interface or circuit type used to create a local switching connection. • *number*—Integer that identifies the number of the interface or circuit.

Configuring CEM-to-CEM ACR Local Switching

Perform this task to configure ACR for CEM-to-CEM local switching.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller sonet** *slot/subslot/port*
4. **aps group** [**acr**] *group-number*
5. **aps working** *circuit-number*
6. **aps protect** *circuit-number ip-address*
7. **exit**
8. **controller sonet-acr** *acr-group-number*
9. **framing sonet**
10. **sts-1** *number*
11. **mode vt-15**
12. **vtg** *number* **t1** *number* **cem-group** *number* **timeslots** *number*
13. **exit**
14. **exit**
15. **interface cem-acr** *acr-group-number*
16. **exit**
17. **cem** *slot/port/channel*
18. **xconnect** *virtual-connect-id*
19. **exit**
20. **exit**
21. **connect** *connection-name type number circuit-id type number circuit-id*

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	**controller sonet** *slot/subslot/port* **Example:** `Router(config)# controller sonet 8/0/0`	Specifies a virtual controller and enters SONET controller configuration mode.
Step 4	**aps group** [**acr**] *group-number*	Configures an ACR working and protect interface.

	Command or Action	Purpose
	Example: `Router(config-controller)# aps group acr 1`	• *group-number*—Number of the group.
Step 5	**aps working** *circuit-number* **Example:** `Router(config-controller)# aps working 1`	Enables a SONET interface as the working interface. • *circuit-number*—Number of the circuit that will be enabled as the working interface. Repeat Steps 3 to 5 for the protect interface.
Step 6	**aps protect** *circuit-number ip-address* **Example:** `Router(config-controller)# aps protect 1 10.0.0.1`	Enables a SONET interface as the protect interface. • *circuit-number*—Number of the circuit that will be enabled as the protect interface. • *ip-address*—IP address of the router that has the working SONET interface.
Step 7	**exit** **Example:** `Router(config-controller)# exit`	Exits SONET controller configuration mode and returns to global configuration mode.
Step 8	**controller sonet-acr** *acr-group-number* **Example:** `Router(config)# controller SONET-acr 1`	Specifies the SONET ACR controller and enters SONET controller configuration mode. • *acr-group-number*—The group number assigned to the working and protect interface.
Step 9	**framing sonet** **Example:** `Router(config-controller)# framing sonet`	Configures the controller framing for SONET framing.
Step 10	**sts-1** *number* **Example:** `Router(config-controller)# sts-1 2`	Specifies the STS identifier and enters STS configuration mode.
Step 11	**mode vt-15** **Example:** `Router(config-ctrlr-sts1)# mode vt-15`	Specifies VT-15 as the STS-1 mode of operation.

	Command or Action	Purpose
Step 12	**vtg** *number* **t1** *number* **cem-group** *number* **timeslots** *number* **Example:** `Router(config-ctrlr-sts1)# vtg 2 t1 4 cem-group 2 timeslots 1-5,14`	Creates a virtual tributary group carrying a single T1 Circuit Emulation Service over Packet Switched Networks (CESoPSN) group.
Step 13	**exit** **Example:** `Router(config-ctrlr-sts1)# exit`	Exits STS configuration mode and returns to SONET controller configuration mode.
Step 14	**exit** **Example:** `Router(config-controller)# exit`	Exits SONET controller configuration mode and returns to global configuration mode.
Step 15	**interface cem-acr** *acr-group-number* **Example:** `Router(config)# interface cem-acr 1`	Specifies the CEM-ACR interface and enters interface configuration mode.
Step 16	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 17	**cem** *slot/port/channel* **Example:** `Router(config)# cem 1/2/0`	Configures CEM and enters circuit emulation (CEM) configuration mode.
Step 18	**xconnect** *virtual-connect-id* **Example:** `Router(config-cem)# xconnect 0`	Builds the CEM connection and enters CEM xconnect configuration mode. • *virtual-connect-id*—Virtual connect ID (VCID).
Step 19	**exit** **Example:** `Router(config-cem-xconnect)# exit`	Exits CEM xconnect configuration mode and returns to CEM configuration mode.
Step 20	**exit** **Example:** `Router(config-cem)# exit`	Exits CEM configuration mode and returns to global configuration mode. Repeat Steps 15 to 19 for the other CEM group.
Step 21	**connect** *connection-name type number circuit-id type number circuit-id*	Defines a connection between the two CEM-ACR circuits. • *connection-name*—Local switching connection name.

	Command or Action	Purpose
	Example: `Router(config)# connect connect1 cem-acr 1 2 cem-acr 2 3`	• *type*—Interface or circuit type used to create a local switching connection. • *number*—Integer that identifies the number of the interface or circuit. • *circuit-id*—CEM group ID.

Verifying Layer 2 Local Switching

Verifying Layer 2 Local Switching Configuration

To verify configuration of the Layer 2 Local Switching feature, use the following commands on the provider edge (PE) router:

SUMMARY STEPS

1. **show connection** [**all** | *element* | **id** *id* | **name** *name* | **port** *port*]
2. **show atm pvc**
3. **show frame-relay pvc** [*pvc*]

DETAILED STEPS

Step 1 **show connection** [**all** | *element* | **id** *id* | **name** *name* | **port** *port*]

The **show connection** command displays the local connection between an ATM interface and a Fast Ethernet interface:

Example:

```
Router# show connection name atm-eth-con
ID  Name            Segment 1              Segment 2         State
=================================================================
1   atm-eth-con     ATM0/0/0 AAL5 0/100    FastEthernet6/0/0  UP
```

This example displays the local connection between an ATM interface and a serial interface:

Example:

```
Router# show connection name atm-fr-con
ID  Name            Segment 1              Segment 2         State
=================================================================
1   atm-fr-con     ATM0/0/0 AAL5 0/100    Serial1/0/0  16     UP
```

This example displays a same-port connection on a serial interface.

Example:

```
Router# show connection name same-port
ID  Name            Segment 1               Segment 2         State
=================================================================
1   same-port      Serial1/1/1  101       Serial1/1/1  102    UP
```

Step 2 **show atm pvc**

The **show atm pvc**command shows that interface ATM3/0 is UP:

Example:

```
Router# show atm pvc
            VCD/                                          Peak   Avg/Min   Burst
Interface   Name   VPI   VCI   Type   Encaps      SC  Kbps   Kbps      Cells    Sts
3/0         10     1     32    PVC    FRATMSRV    UBR 155000                    UP
```

Step 3 **show frame-relay pvc** [*pvc*]

The **show frame-relay pvc** command shows a switched Frame Relay PVC:

Example:

```
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
```

Verifying the NSF SSO Local Switching Configuration

Layer 2 local switching provides NSF/SSO support for Local Switching of the following attachment circuits on the same router:

- Ethernet (port mode) to Ethernet VLAN
- Frame Relay to Frame Relay

For information about configuring NSF/SSO on the RPs, see the Stateful Switchover feature module. To verify that the NSF/SSO: Layer 2 Local Switching is working correctly, follow the steps in this section.

SUMMARY STEPS

1. Issue the **ping**command or initiate traffic between the two CE routers.
2. Force the switchover from the active RP to the standby RP by using the **redundancy force-switchover** command. This manual procedure allows for a "graceful" or controlled shutdown of the active RP and switchover to the standby RP. This graceful shutdown allows critical cleanup to occur.
3. Issue the **show connect all** command to ensure that the Layer 2 local switching connection on the dual RP is operating.
4. Issue the **ping** command from the CE router to verify that the contiguous packet outage was minimal during the switchover.

DETAILED STEPS

Step 1 Issue the **ping**command or initiate traffic between the two CE routers.

Step 2 Force the switchover from the active RP to the standby RP by using the **redundancy force-switchover** command. This manual procedure allows for a "graceful" or controlled shutdown of the active RP and switchover to the standby RP. This graceful shutdown allows critical cleanup to occur.

Step 3 Issue the **show connect all** command to ensure that the Layer 2 local switching connection on the dual RP is operating.

Example:

```
Router# show connect all
ID      Name          Segment 1     Segment 2     State
2      Eth-Vlan1      Fa1/1/1       Fa6/0/0/0.1   UP
```

Step 4 Issue the **ping** command from the CE router to verify that the contiguous packet outage was minimal during the switchover.

Troubleshooting Tips

You can troubleshoot Layer 2 local switching using the following commands on the PE router:

- **debug atm l2transport**
- **debug conn**
- **debug frame-relay pseudowire**
- **show atm pvc**
- **show connection**
- **show frame-relay pvc**

Configuration Examples for Layer 2 Local Switching

Example: Configuring ATM-to-ATM Local Switching

The following example shows local switching on ATM interfaces configured for AAL5:

```
interface atm1/0/0
 pvc 0/100 l2transport
 encapsulation aal5
interface atm2/0/0
 pvc 0/100 l2transport
 encapsulation aal5
connect aal5-conn atm1/0/0 0/100 atm2/0/0 0/100
```

Example: Configuring ATM PVC Same-Port Switching

The following example shows same-port switching between two PVCs on one ATM interface:

```
interface atm1/0/0
 pvc 0/100 l2transport
 encapsulation aal5
 pvc 0/200 l2transport
 encapsulation aal5
connect conn atm1/0/0 0/100 atm1/0/0 0/200
```

Example: Configuring ATM PVP Same-Port Switching

The following example shows same-port switching between two PVPs on one ATM interface:

```
interface atm1/0/0
 atm pvp 100 l2transport
 atm pvp 200 l2transport
connect conn atm1/0/0 100 atm1/0/0 200
```

Example: ATM-to-Ethernet Local Switching

ATM-to-Ethernet local switching terminates an ATM frame to an Ethernet/VLAN frame over the same PE router. Two interworking models are used: Ethernet mode and IP mode.

Example: ATM-to-Ethernet VLAN Mode Local Switching

The following example shows an Ethernet interface configured for Ethernet VLAN, and an ATM PVC interface configured for AAL5 encapsulation. The **connect** command allows local switching between these two interfaces and specifies the interworking type as Ethernet mode.

Note On the provider edge (PE) router, ensure that the maximum transmission unit (MTU) value of ATM (default MTU is 4470 bytes) and GigabitEthernet (default MTU is 1500 bytes) interfaces is the same. On the customer edge (CE) router, ensure that the MTU value of ATM and GigabitEthernet interfaces is at least 14 bytes less than the MTU value of the respective interfaces on the PE router during ATM-to-Ethernet VLAN mode local switching.

```
interface fastethernet6/0/0.1
 encapsulation dot1q 10
interface atm2/0/0
 pvc 0/400 l2transport
 encapsulation aal5
connect atm-ethvlan-con atm2/0/0 0/400 fastethernet6/0/0.1 interworking ethernet
```

Example: ATM-to-Ethernet Port Mode Local Switching

The following example shows an Ethernet interface configured for Ethernet and an ATM interface configured for AAL5SNAP encapsulation. The **connect** command allows local switching between these two interfaces and specifies the interworking type as IP mode.

```
interface atm0/0/0
 pvc 0/100 l2transport
 encapsulation aal5snap
interface fastethernet6/0/0
connect atm-eth-con atm0/0/0 0/100 fastethernet6/0/0 interworking ip
```

Example: Ethernet VLAN Same-Port Switching

The following example shows same-port switching between two VLANs on one Ethernet interface:

```
interface fastethernet0/0.1
 encapsulation dot1q 1
interface fastethernet0/0.2
 encapsulation dot1q 2
connect conn FastEthernet0/0.1 FastEthernet0/0.2
```

Example: ATM-to-Frame Relay Local Switching

The following example shows a serial interface configured for Frame Relay and an ATM interface configured for AAL5SNAP encapsulation. The **connect** command allows local switching between these two interfaces.

```
interface serial1/0
 encapsulation frame-relay
interface atm1/0
 pvc 7/100 l2transport
 encapsulation aal5snap
connect atm-fr-conn atm1/0 7/100 serial1/0 100 interworking ip
```

Example: Frame Relay-to-Frame Relay Local Switching

The following example shows serial interfaces configured for Frame Relay. The **connect** command allows local switching between these two interfaces.

```
frame-relay switching
ip cef distributed
interface serial3/0/0
```

```
 encapsulation frame-relay
 frame-relay interface-dlci 100 switched
 frame-relay intf-type dce
interface serial3/1/0
 encapsulation frame-relay ietf
 frame-relay interface-dlci 200 switched
 frame-relay intf-type dce
connect fr-con serial3/0/0 100 serial3/1/0 200
```

Example: Frame Relay DLCI Same-Port Switching

The following example shows same-port switching between two data links on one Frame Relay interface:

```
interface serial1/0
 encapsulation frame-relay
 frame-relay int-type nni
connect conn serial1/0 100 serial1/0 200
```

Example: HDLC Local Switching

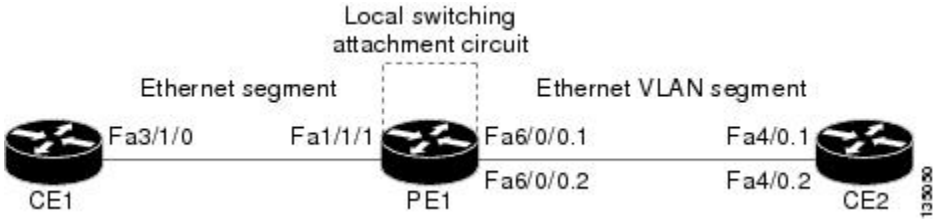
The following example shows local switching of two serial interfaces for HDLC:

```
interface serial1/0
 no ip address
interface serial2/0
 no ip address
connect conn1 serial1/0 serial1/0
```

Example: NSF SSO Ethernet Port Mode to Ethernet VLAN Local Switching

The following configuration uses the network topology shown in the figure below.

Figure 12: NSF/SSO: Layer 2 Local Switching: Ethernet to Ethernet VLAN



The following example shows the configuration of the CE interfaces to connect to the PE1 router:

CE1	CE2
ip routing ! interface fa3/1/0 description: connection to PE fa1/1/1 no shutdown ip address 10.1.1.1 255.255.255.0	ip routing ! interface fa4/0 no shutdown ! interface fa4/0.1 description: connection to PE1 fa6/0/0.1 encapsulation dot1Q 10 ip address 10.1.1.2 255.255.255.0 ! interface fa4/0.2 description - connection to PE1 fa6/0/0.2 encapsulation dot1Q 20 ip address 172.16.1.2 255.255.255.0

The following example shows the configuration of the PE1 router with NSF/SSO and the PE interfaces to the CE routers:

PE1

```
redundancy
 no keepalive-enable
 mode sso
!
hw-module slot 2 image disk0:rsp-pv-mz.shaft.111004
hw-module slot 3 image disk0:rsp-pv-mz.shaft.111004
!
ip routing
ip cef distributed
!
interface fa1/1/1
 description - connection to CE1 fa3/1/0
 no shutdown
 no ip address
!
interface fa4/0/0
 description - connection to CE3 fa6/0
 no shutdown
 no ip address
!
interface fa6/0/0
 no shutdown
 no ip address
!
interface fa6/0/0.1
 description - connection to CE2 fa4/0.1
 encapsulation dot1Q 10
 no ip address
!
interface fa6/0/0.2
 description - connection to CE2 fa4/0.2
 encapsulation dot1Q 20
 no ip address
```

The following example shows the configuration of ICMP Router Discovery Protocol (IRDP) on the CE router for Interworking IP for ARP mediation:

CE1	CE2

`interface FastEthernet3/1/0`
` ip irdp`
` ip irdp maxadvertinterval 0`
	`interface FastEthernet4/0.1`
	` ip irdp`
	` ip irdp maxadvertinterval 0`

The following example shows the configuration of OSPF on the CE routers:

CE1	CE2

`interface loopback 1`
` ip address 10.11.11.11 255.255.255.255`
`!`
`router ospf 10`
` network 10.11.11.11 0.0.0.0 area 0`
` network 192.168.1.1 0.0.0.0 area 0`
	`interface loopback 1`
	` ip address 12.12.12.12 255.255.255.255`
	`!`
	`router ospf 10`
	` network 10.12.12.12 0.0.0.0 area 0`
	` network 192.168.1.2 0.0.0.0 area 0`

The following example shows the configuration of local switching on the PE1 router for interworking Ethernet:

```
connect eth-vlan1 fa1/1/1 fa6/0/0.1 interworking ethernet
connect eth-vlan2 fa4/0/0 fa6/0/0.2 interworking ethernet
```

The following example shows the configuration of local switching on the PE1 router for interworking IP:

```
connect eth-vlan1 fa1/1/1 fa6/0/0.1 interworking ip
connect eth-vlan2 fa4/0/0 fa6/0/0.2 interworking ip
```

Additional References for Layer 2 Local Switching

Related Documents

Related Topic	Document Title
MPLS	*MPLS Product Literature*
Layer 2 local switching configuration tasks	Configuring Layer 2 Local Switching
Frame Relay-ATM interworking configuration tasks	Configuring Frame Relay-ATM Interworking
Frame Relay-to-Frame Relay local switching configuration tasks	Distributed Frame Relay Switching
CEoP and Channelized ATM SPAs on Cisco 7600 series router configuration tasks	Configuring the CEoP and Channelized ATM SPAs

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*
draft-martini-l2circuit-encap-mpls-04.txt	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-ppvpn-l2vpn-00.txt	*An Architecture for L2VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Layer 2 Local Switching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

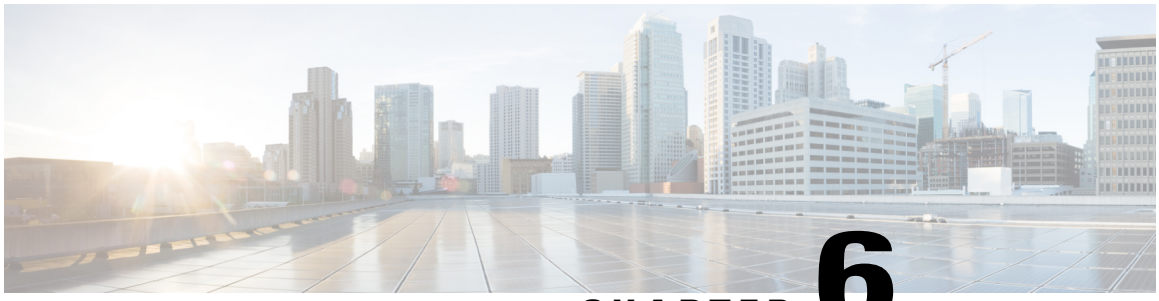
Table 17: Feature Information for Layer 2 Local Switching

Feature Name	Releases	Feature Information
Layer 2 Local Switching	12.0(27)S 12.0(30)S 12.0(31)S2 12.0(32)SY 12.2(25)S 12.2(28)SB 12.2(33)SB 12.2(33)SRB 12.2(33)SXH 12.4(11)T 15.0(1)S	

Feature Name	Releases	Feature Information
		The Layer 2 Local Switching feature allows you to switch Layer 2 data between two interfaces on the same router, and in some cases to switch Layer 2 data between two circuits on the same interface port. • The feature was introduced in Cisco IOS Release 12.0(27)S on the Cisco 7200 and 7500 series routers. • In Cisco IOS Release 12.0(30)S, support for same-port switching was added. Support for Layer 2 interface-to-interface local switching was added on the GSR. • In Cisco IOS Release 12.0(31)S2, support was added for customer edge-facing IP Service Engine (ISE) interfaces on the GSR. • In Cisco IOS Release 12.0(32)SY, support was added for customer edge-facing interfaces on Engine 5 shared port adapters (SPAs) and SPA Interface Processors (SIPs) on the GSR. • The feature was integrated into Cisco IOS Release 12.2(25)S for the Cisco 7500 series router. • In Cisco IOS Release 12.2(28)SB, this feature was updated to include NSF/SSO support on the Cisco 7500 series routers for the following local switching types on nonstop forwarding/stateful switchover (NSF/SSO): • NSF/SSO—Ethernet-to-Ethernet

Feature Name	Releases	Feature Information
		VLAN local switching support • NSF/SSO—Frame Relay-to-Frame Relay local switching support • In Cisco IOS Release 12.2(28)SB, support was added for Local Switching on the Cisco 10000 series router. • In Cisco IOS Release 12.2(33)SB, support was added for HDLC Local Switching on the Cisco 7200 series router and the Cisco 10000 series router. • In Cisco IOS Release 12.2(33)SXH, support was added for like-to-like Local Switching (ATM to ATM, and FR to FR only) on Cisco 6500 series switches and Cisco 7600 series routers. Same-port switching is not supported on those routers. • In Cisco IOS Release 12.4(11)T, support was added for the following local switching types for the Cisco 7200 series router: • Ethernet to Ethernet VLAN • Same-port switching for Ethernet VLAN • Frame Relay to Frame Relay • Same-port switching for Frame Relay The following commands were introduced or modified: **connect** (L2VPN local switching), **encapsulation** (Layer 2 local switching), **show connection**.

Feature Name	Releases	Feature Information
Access Circuit Redundancy for ATM Local Switching	15.1(1)S	Access Circuit Redundancy (ACR) ensures low data traffic downtime by reducing the switchover time. ACR works on the APS 1+1, nonrevertive model where each redundant line pair consists of a working line and a protect line. If a signal fail condition or a signal degrade condition is detected, the hardware switches from the working line to the protect line. In Cisco IOS Release 15.1(1)S, this feature was introduced. The following commands were introduced or modified: **aps group**, **connect** (L2VPN local switching).
ACR Support for CEM	15.1(1)S	This feature provides Access Circuit Redundancy (ACR) support for CEM. In Cisco IOS Release 15.1(1)S, this feature was introduced. The following commands were introduced or modified: **aps group**, **connect** (L2VPN local switching).



CHAPTER 6

Stateful MLPPP with MR-APS

First Published: July 22, 2011

Last Updated: July 22, 2011

The Stateful MLPPP with MR-APS feature supports Interchassis Stateful Switchover (IC-SSO) for Multilink PPP (MLPPP) sessions, thereby allowing Multirouter Automatic Protection Switching (MR-APS) from one router to another, while maintaining the MLPPP sessions and avoiding session renegotiation. This feature is available only on Cisco 7600 series routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information for Stateful MLPPP with MR-APS, on page 246.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Prerequisites for Configuring Stateful MLPPP with MR-APS

- To enable Stateful MLPPP with MR-APS across two routers, both routers must be manually configured with similar MR-APS MLPPP configurations.
- SONET controllers must be configured and enabled on the routers before the Stateful MLPPP with MR-APS feature can be configured.

Restrictions for Stateful MLPPP with MR-APS

- In-Service Software Upgrade (ISSU) is not supported.
- Applications running over PPP/MLPPP sessions such as Internet Group Management Protocol (IGMP) and TCP are not synchronized across the chassis. During Automatic Protection Switchover (APS), IGMP joints and TCP sessions need to be reestablished.
- APS session throttling for groups is not supported.
- Broadband sessions such as Point-to-Point Protocol over X (PPPoX) and IP are not supported in this feature.
- Intelligent Services Gateway (ISG) features are not supported on APS groups.
- The Authentication, Authorization, and Accounting (AAA) protocol is not supported on MR-APS.
- Config-sync is not supported.
- To enable Stateful MLPPP with MR-APS across two routers, both routers must be manually configured with similar MR-APS MLPPP configurations.

Information About Stateful MLPPP with MR-APS

Stateful MLPPP with MR-APS Overview

Traditionally, Multirouter Automatic Protection Switching provides Layer 1 (L1) switchover for optical links under 50 milliseconds across two routers. However, if there are MLPPP or PPP sessions on the optical link during an MR-APS switchover, all active MLPPP or PPP sessions need to renegotiate resulting in traffic loss.

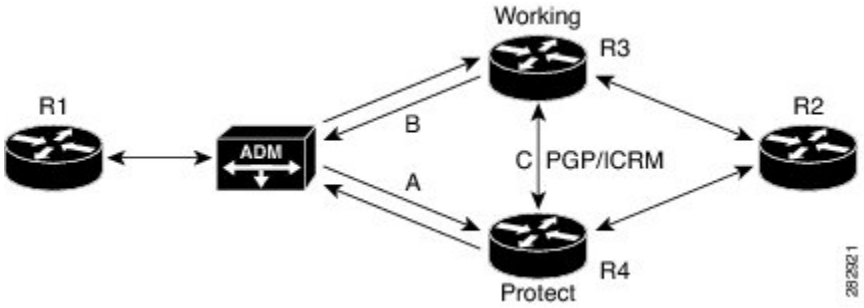
The Stateful MLPPP with MR-APS feature provides IC-SSO for PPP and MLPPP sessions across two routers without renegotiating the session or reprogramming the hardware when the switchover occurs. IC-SSO for MLPPP maintains the control plane state by synchronizing it from the router hosting the MR-APS active interface to the router hosting the MR-APS inactive interface. Using this synchronized information, the second router maintains the forwarding plane in a state of readiness to forward traffic immediately after an MR-APS switchover.

Interchassis MR-APS MLPPP SSO is achieved by leveraging and enhancing the existing functionality of MR-APS, Interchassis Redundancy Manager (ICRM), MLPPP, and Cluster Control Manager (CCM) components and protocols.

MR-APS Deployment

The MR-APS deployment involves multiple cell site routers connected to the provider network using bundled T1/E1 connections. These T1/E1 connections are aggregated into Optical Carrier 3 (OC3) or Optical Carrier 12 (OC12) links using Add-Drop Multiplexers (ADMs). The figure below shows the MR-APS deployment using Cisco 7600 routers. Router 1 (R1) is the cell site router, Router 2 (R2) is the core router, Routers 3 (R3) is the working provider edge (PE) router, and Router 4 (R4) is the protect PE router. To implement the Stateful MLPPP with MR-APS feature, you must configure MR-APS IC-SSO on both the working and the protect Cisco 7600 series routers.

Figure 13: MR-APS Deployment



Unlike the conventional SSO model, where one router is active and the other is in standby mode, in IC-SSO, during an MR-APS deployment, both routers are in the active state with SONET controllers synchronized on both routers. The controllers running on one router are in standby mode on the other router and vice versa. When MR-APS detects a failure in a SONET OC3 or OC12 controller on the working router, it activates the corresponding inactive controller on the protect router. This switchover from the inactive to the active state ensures minimal traffic outage to the end user, and this is achieved by ensuring that the MLPPP/PPP sessions per SONET controller (APS group) are stateful across the routers.

Interchassis Redundancy Manager

The Interchassis Redundancy Manager (ICRM) provides the following capabilities for the implementation of the Stateful MLPPP with MR-APS feature:

Node-health monitoring for complete node/PE/box failure detection. ICRM also detects failures to applications registered with an ICRM group.

Reliable data channeling to transfer state information to the peer.

Active RP failure detection. This failure is detected as a node failure and the controllers are notified.

- On failure of the active Route Processor (RP), ICRM on the standby RP reestablishes the communication channel with the peer node.

Automatic Protection Switching

Automatic Protection Switching (APS), the building block of the MR-APS feature, is responsible for managing the active and standby progression events on APS groups. Each APS group is a logical representation of a physical SONET controller redundancy state.

APS allows the switchover of OC3/OC12 channels in the event of a failure. APS involves a protect interface in the network as the backup for an active (working) interface. When the active interface fails, the protect interface takes care of the traffic load. Depending on the configuration, the two interfaces may be terminated on the same router or different routers. Based on where the interfaces terminate, APS is categorized into two types: single-router APS (SR-APS) and multirouter APS (MR-APS).

CCM Enhancements

The Cluster Control Manager (CCM) acts as a high availability (HA) abstract layer for all types of PPP sessions. The CCM is responsible for collecting all the required information from all clients that are part of a given session and syncing the information to the standby RP, thereby re-creating the session on the standby RP. Traditionally, the CCM is only aware of the RP HA state, which is either standby or active. This means that if the RP is active, the CCM treats all sessions on that RP as active, and if the RP is standby, the CCM treats all sessions on that RP as standby.

However, for the implementation of the Stateful MLPPP with MR-APS feature, the CCM is enhanced to have logical partitions of CCM sessions, also known as CCM groups. These CCM groups provide the capability to logically group broadband sessions and apply redundancy operations to only those set of sessions that belong to a CCM group. This feature enables broadband routers to act as standby for a group of broadband sessions that are active on a remote router, while hosting its own active broadband sessions. Therefore, this enhancement will enable each CCM group to be either active or standby on a given active RP and a given active RP to have multiple active CCM groups and multiple standby CCM groups.

Redundancy Group Facility

A new module called the redundancy group facility (RGF) has been developed to act as an agent between CCM, ICRM, and APS. This module is responsible for propagating redundancy state progressions to the CCM by receiving the redundancy state as active or standby from APS and deriving the CCM group progressions to reach either the active or the standby hot state. RGF also works as a mediator between ICRM and CCM groups for check-pointing session data. It will also accept node failure events from ICRM and propagate them to CCM groups.

Failure Protection Scenarios

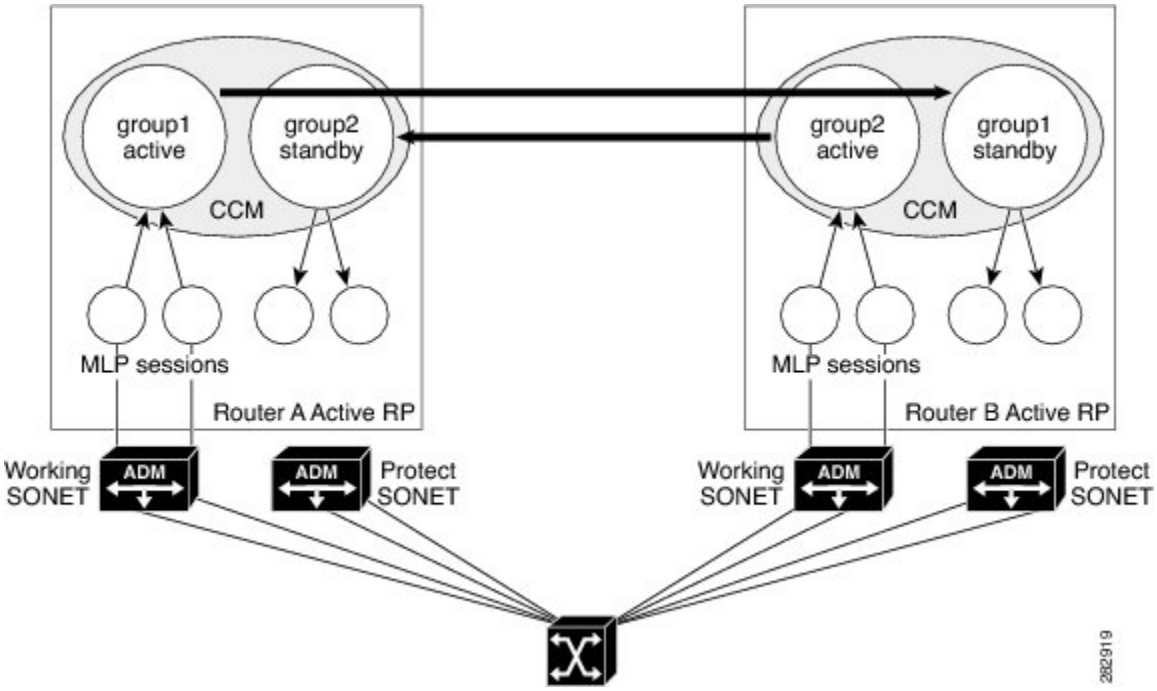
The Stateful MLPPP feature provides network resiliency by protecting against the following scenarios:

Active APS SONET Controller Failure

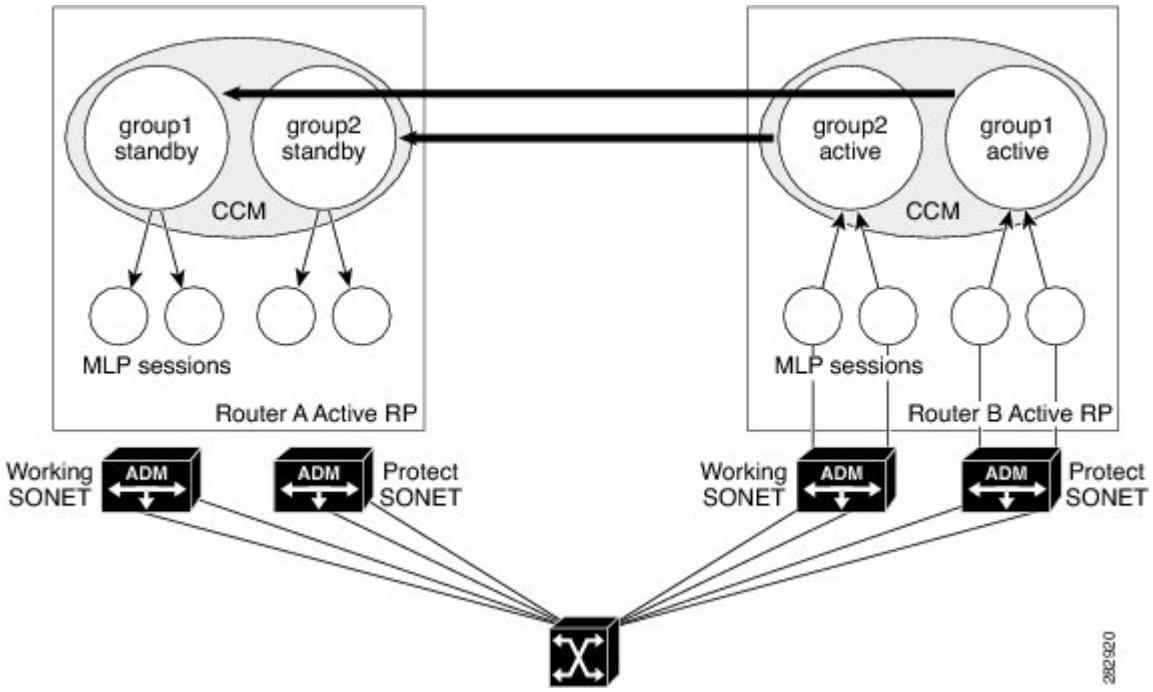
The figure below shows MLPPP sessions in an MR-APS configuration before an active APS group fails. On Router A active RP, group1 is CCM group 1 and group2 is CCM group 2. All sessions of group1 are active

and all sessions of group2 are standby on Router A. Similarly, on Router B, all sessions of group2 are active and all sessions of group1 are in standby state.

Figure 14: MLPPP Sessions Before an Active APS Group Fails



When an APS group on Router A fails, the APS informs the corresponding standby APS group on Router B to take over as the active APS group. Here APS will be enhanced to inform CCM about the failure to the corresponding CCM group. The CCM group takes over as the active group and all sessions in that group will become active, while the previous active CCM group reinitializes itself before moving into the standby state. The figure below shows how MLPPP sessions switch over after the failure of an active APS group.



The standby group1 on the remote router takes over as the active group and reinitializes itself before going into the standby state.

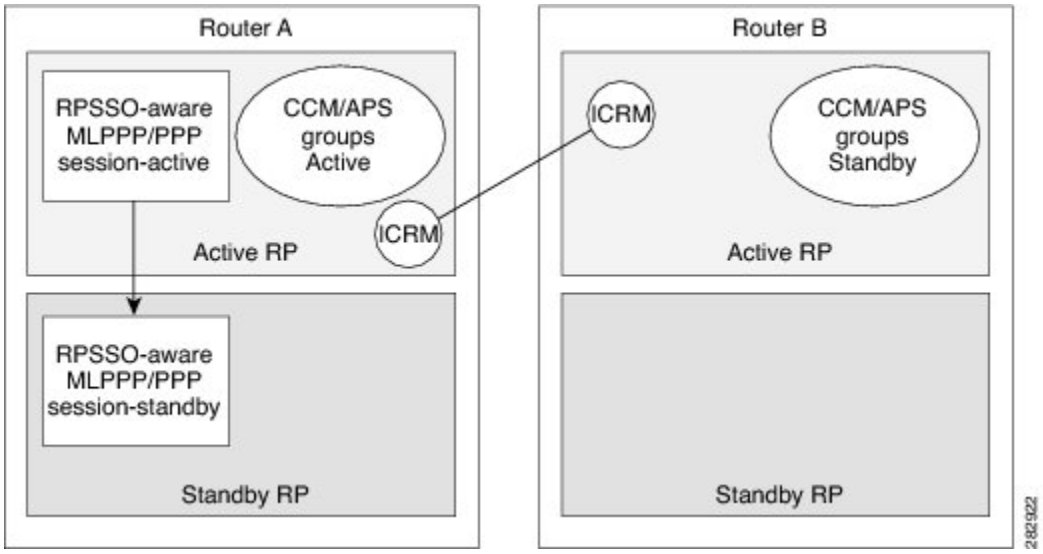
RP Failure and Node Failure

ICRM treats an active RP failure as a complete node failure and sends the go-active event to all standby CCM groups directing them to take over as active. Also, all standby APS groups move to active state on receiving the go-active event message, ensuring that both the APS and CCM groups are in the same state, even though APS can detect node failure on its own. Standby CCM groups take over as active and RGF updates its groups with the “peer not available” status.

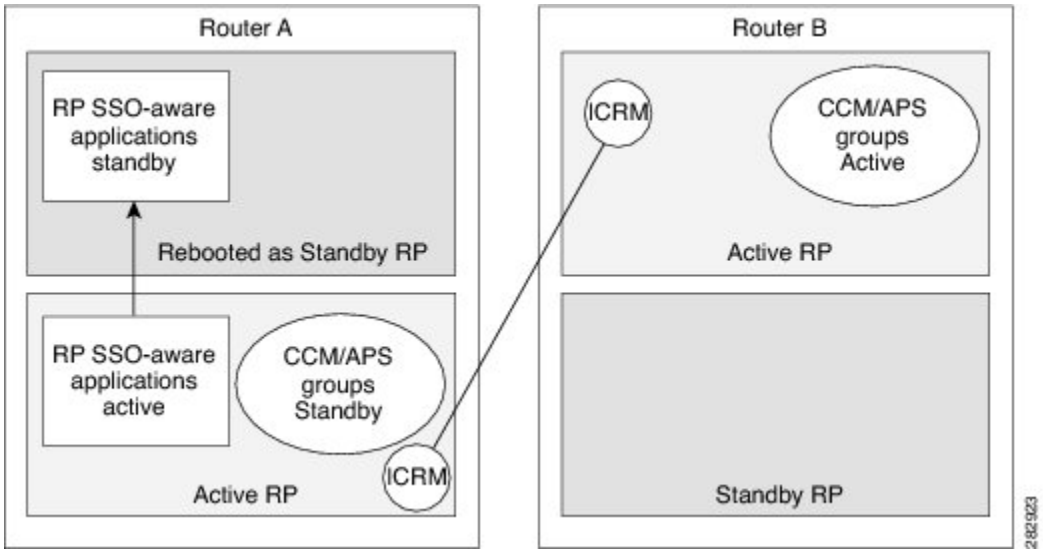
When the failed node comes up, ICRM establishes fresh connectivity and RGF connects to all groups on the remote router that is becoming active. Since peer groups are detected, RGF ensures bulk syncing of active CCM groups. The standby groups on the peer box receive this bulk sync data and automatically move into a hot-standby state.

The figure below shows CCM/APS groups on two peer nodes: Router A and Router B.

Figure 15: APS Groups on Peer Nodes



When the active RP of Router A fails, applications using ICRM should switch over to Router B (remote box). Consequently, all APS/CCM groups should switch over to Router B. Now, Router B has all the active APS/CCM groups. All APS/CCM groups on the standby RP of Router A are set to Init state after the standby RP changes to the active RP on Router A. Applications that are RP SSO aware (non-ICRM clients) switch over to the standby RP on Router A. The figure below shows APS groups after the active RP on Router A fails.



The ICRM establishes fresh connections with the new active RP on Router A and APS synchronizes the group states from Router B to Router A in the standby state. This event triggers all APS groups on Router A to go to the standby state, and the synchronization process is initiated from Router B. On Router A, the failed RP reboots as the new standby RP and RP SSO-aware applications are synchronized to the new standby RP.

How to Configure Stateful MLPPP with MR-APS

Setting Up an ICRM Session

Perform this task on both the working and the protect router to set up ICRM sessions to establish communication between the routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **interchassis group** *group-id*
5. **monitor peer bfd**
6. **member ip** *ip-address*
7. **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	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 50`	Configures an interchassis group within redundancy configuration mode and enters interchassis redundancy mode.
Step 5	**monitor peer bfd**	Configures the BFD option to monitor the state of the peer.

	Command or Action	Purpose
	Example: `Router(config-r-ic)# monitor peer bfd`	• The default configuration is route-watch.
Step 6	**member ip** *ip-address* **Example:** `Router(config-r-ic)# member ip 10.60.60.1`	Configures a remote redundancy group member by specifying the IP address of the member.
Step 7	**end** **Example:** `Router(config-r-ic)# end`	Exits interchassis redundancy mode and returns to privileged EXEC mode.

Setting Up the BFD Interval

Perform this task on both the working and the protect router to set up the baseline Bidirectional Forwarding Detection (BFD) parameters between the routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subplot* / *port*
4. **ip address** *ip-address subnet-mask*
5. **load-interval** *seconds*
6. **negotiation** {**forced**| **auto**}
7. **mpls ip**
8. **mpls label protocol** {**ldp** | **tdp** | **both**}
9. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subplot* / *port* **Example:** `Router(config)# interface GigabitEthernet3/1/0`	Specifies the Gigabit Ethernet interface to be configured, where *slot/subslot/port* specifies the location of the interface.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Configures the IP address for the interface.
Step 5	**load-interval** *seconds* **Example:** `Router(config-if)# load-interval 30`	Sets the length of time for which data is used for load calculations.
Step 6	**negotiation** {**forced**| **auto**} **Example:** `Router(config-if)# negotiation auto`	Enables the negotiation of speed, duplex mode, and flow control on a Gigabit Ethernet interface.
Step 7	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS.
Step 8	**mpls label protocol** {**ldp** | **tdp** | **both**} **Example:** `Router(config-if)# mpls label protocol both`	Configures the label or tag distribution protocol or both on the interface.
Step 9	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*	Configures the transmit interval between BFD packets.

	Command or Action	Purpose
	Example: `Router(config-if)# bfd interval 50 min_rx 150 multiplier 3`	
Step 10	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the SONET Controller

Perform this task on the working and the protect router to configure SONET controllers on the routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller sonet** *slot* **/** *bay* **/** *port*
4. **no ais-shut**
5. **framing sonet**
6. **clock source** {**line** | **interval**}
7. **sts-1** *sts1-number*
8. **mode vt-15**
9. **vtg** *vtg-number* **t1** *t1-line-number* **channel-group** *channel-number* **timeslots** *list-of-timeslots*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller sonet** *slot* / *bay* / *port* **Example:** `Router(config)# controller sonet 3/2/0`	Selects and configures a SONET controller and enters controller configuration mode.
Step 4	**no ais-shut** **Example:** `Router(config-controller)# no ais-shut`	Disables automatic insertion of a line alarm indication signal (LAIS) in the SONET signal.
Step 5	**framing sonet** **Example:** `Router(config-controller)# framing sonet`	Configures the controller for SONET framing; SONET framing is the default configuration.
Step 6	**clock source** {**line** | **interval**} **Example:** `Router(config-controller)# clock source line`	Configures the SONET port transmit clock source, where the **internal** keyword sets the internal clock and **line** keyword sets the clock recovered from the line. • Use the **line** keyword whenever clocking is derived from the network. Use the **internal** keyword when two routers are connected back-to-back or over fiber for which no clocking is available. • The line clock is the default configuration.
Step 7	**sts-1** *sts1-number* **Example:** `Router(config-controller)# sts-1 1`	Specifies the Synchronous Transport Signal (STS) identifier and enters STS configuration mode.
Step 8	**mode vt-15** **Example:** `Router(config-ctrlr-sts1)# mode vt-15`	Specifies VT-15 as the STS-1 mode of operation.
Step 9	**vtg** *vtg-number* **t1** *t1-line-number* **channel-group** *channel-number* **timeslots** *list-of-timeslots*	Creates a Circuit Emulation Services over Packet Switched Network (CESoPSN) circuit emulation CEM group.

	Command or Action	Purpose
	Example: `Router(config-ctrlr-sts1)# vtg 1 t1 1 channel-group 0 timeslots 1-24`	
Step 10	**end**	Exits STS configuration mode and returns to privileged EXEC mode.

Configuring the Serial Interface to Enable MLPPP

Perform this task on both the working and the protect router to configure the serial interface to enable MLPPP sessions on the routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *instance*
4. **no ip address**
5. **encapsulation ppp**
6. **ppp multilink**
7. **ppp multilink group** *group-number*
8. **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.

	Command or Action	Purpose
Step 3	**interface serial** *instance* **Example:** `Router(config)# interface Serial3/2/0.1/1/1:0`	Configures the serial interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes any configured IP address from the interface.
Step 5	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation of traffic on the specified interface.
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLPPP.
Step 7	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink group 1`	Restricts a physical link to be associated only with a designated multilink group interface.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the Multilink Interface

Perform this task on both the working and the protect router to configure the multilink interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink1**
4. **ip address** *ip-address subnet-mask*
5. **carrier-delay msec** *msec*
6. **ppp multilink**
7. **ppp multilink group** *group-number*
8. **ppp multilink endpoint** {**hostname** | **ip** *ip-address* | **mac** *lan-interface* | **none** | **phone** *telephone-number* | **string** *char-string*}
9. **ppp timeout retry** *seconds*
10. **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	**interface multilink1** **Example:** `Router(config)# interface multilink1`	Configures a multilink interface and enters multilink interface configuration mode.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Configures the IP address for the interface.
Step 5	**carrier-delay msec** *msec* **Example:** `Router(config-if)# carrier-delay msec 1`	Sets the time to propagate the link status to other modules.

	Command or Action	Purpose
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLPPP.
Step 7	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink group 1`	Restricts a physical link to be associated only with a designated multilink group interface.
Step 8	**ppp multilink endpoint** {**hostname** | **ip** *ip-address* | **mac** *lan-interface* | **none** | **phone** *telephone-number* | **string** *char-string*} **Example:** `Router(config-if)# ppp multilink endpoint string mlp_aps_1`	Overrides or changes the default endpoint discriminator that the system uses when negotiating the use of MLPPP with the peer system. • **hostname** – Specifies the use of the hostname configured for the router. This is useful when multiple routers use the same username for authentication, but have different hostnames. • **ip** *ip-address* – Specifies the IP address to be used. • **mac** *lan-interface* – Specifies the 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** – Specifies the telephone number to be used. Accepts E.164-compliant and full international telephone numbers. • **string** *char-string* – Specifies the specific character string to be used.
Step 9	**ppp timeout retry** *seconds* **Example:** `Router(config-if)# ppp timeout retry 4`	Sets PPP timeout retry parameters. • Specifies the maximum time, in seconds, to wait for a response during PPP negotiation. The range is from 1 to 10 seconds. • The default is 3 seconds.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the APS Group for the SONET Controller

Perform this task on both the working and protect router to configure the APS group for a SONET controller.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller sonet** *slot* / *bay* / *port*
4. **shutdown**
5. **aps group** *group-id*
6. **aps** [**working**| **protect**] *aps-group-number*[*ip-address-working-router*]
7. **aps interchassis group** *group-number*
8. **no shutdown**
9. **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	**controller sonet** *slot* / *bay* / *port* **Example:** `Router(config)# controller sonet 3/2/0`	Selects and configures a SONET controller and enters controller configuration mode.
Step 4	**shutdown** **Example:** `Router(config-controller)# shutdown`	Shuts down the SONET controller.
Step 5	**aps group** *group-id* **Example:** `Router(config-controller)# aps group 1`	Configures an APS group for the SONET controller.

	Command or Action	Purpose
Step 6	**aps** [**working**| **protect**] *aps-group-number*[*ip-address-working-router*] **Example:** `Router(config-controller)# aps working 1 10.2.2.1`	Configures the APS group as the working or protect interface, depending on whether the router is the working router or the protect router. The *ip-address-working-router* attribute is only required while configuring the protect router.
Step 7	**aps interchassis group** *group-number* **Example:** `Router(config-controller)# aps interchassis group 1`	Associates an APS group with an ICRM group number.
Step 8	**no shutdown** **Example:** `Router(config-controller)# no shutdown`	Enables the interface.
Step 9	**end** **Example:** `Router(config-controller)# end`	Exits controller configuration mode and returns to privileged EXEC mode.

Verifying the Functionality of Stateful MLPPP with MR-APS

Perform the following steps to verify the functionality of the Stateful MLPPP with MR-APS feature configured on the working and protect router.

SUMMARY STEPS

1. **show aps**
2. **show rgf groups**

DETAILED STEPS

Step 1 **show aps**

Use this command to display detailed information about the APS configuration on the working or protect router. The following is sample output of the command on the protect router:

Example:

```
Router# show aps
SONET 3/2/0 APS Group 1: protect channel 0 (Inactive) (HA)
Working channel 1 at 10.1.1.2 (Enabled) (HA)
bidirectional, non-revertive
PGP timers (extended for HA): hello time=1; hold time=10
hello fail revert time=120
SONET framing; SONET APS signalling by default
Received K1K2: 0x00 0x05
No Request (Null)
Transmitted K1K2: 0x00 0x05
No Request (Null)
Remote APS configuration: (null)
Protect-Router#
```

The following is sample output of the command on the working router:

Example:

```
Router# show aps
SONET 1/2/0 APS Group 1: working channel 1 (Active) (HA)
Protect at 10.1.1.2
PGP timers (from protect): hello time=1; hold time=10
SONET framing
Remote APS configuration: (null)
```

Step 2 **show rgf groups**

Use this command to get information about the state of the router and the peer. The following is sample output of the command on the protect router:

Example:

```
Router# show rgf groups
Total RGF groups: 1
------------------------------------------------------------
STANDBY RGF GROUP
RGF Group ID : 1
RGF Peer Group ID: 0
ICRM Group ID : 1
APS Group ID : 1
RGF State information:
My State Present : Standby-hot
Previous : Standby-bulk
Peer State Present: Active-fast
Previous: Standby-cold
Misc:
Communication state Up
aps_bulk: 0
aps_stby: 0
peer_stby: 0
-> Driven Peer to [peer Standby Bulk] Progression
-> We sent Bulk Sync start Progression to Active
RGF GET BUF: 366 RGF RET BUF 366
```

The following is sample output of the command on the working router:

Example:

```
Router# show rgf groups
Total RGF groups: 1
------------------------------------------------------------
```

```
ACTIVE RGF GROUP
RGF Group ID : 1
RGF Peer Group ID: 0
ICRM Group ID : 1
APS Group ID : 1
RGF State information:
My State Present : Active-fast
Previous : Standby-cold
Peer State Present: Standby-hot
Previous: Standby-bulk
Misc:
Communication state Up
aps_bulk: 0
aps_stby: 0
peer_stby: 0
-> Driven Peer to [Peer Standby Hot] Progression
-> Standby sent Bulk Sync start Progression
```

RGF GET BUF: 366 RGF RET BUF 366

If the value of “My State Present” is “Standby-hot,” the router is in standby state. If the value of “My State Present” is “Active-fast,” the router is in active state.

Configuration Examples for Stateful MLPPP with MR-APS

This section provides the following configuration examples:

Example Configuring Stateful MLPPP with MR-APS on a Working Router, on page 242

Example Configuring Stateful MLPPP with MR-APS on a Protect Router, on page 243

Example Configuring Stateful MLPPP with MR-APS on a Working Router

This example shows how to configure Stateful MLPPP with MR-APS on a Working Router.

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# interchassis group 1
Router(config-r-ic)# monitor peer bfd
Router(config-r-ic)# member ip 10.1.1.2
Router(config-r-ic)# end
Router#
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface GigabitEthernet3/1/0 <<<<<<< ICRM link >>>>>>>>
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# load-interval 30
Router(config-if)# negotiation auto
Router(config-if)# mpls ip
Router(config-if)# mpls label protocol both
Router(config-if)# bfd interval 50 min_rx 150 multiplier 3
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config-if)# interface GigabitEthernet3/1/1 <<<<< PGP Link>>>>>>>
Router(config-if)# ip address 10.1.1.3 255.255.255.0
Router(config-if)# negotiation auto
Router(config-if)# cdp enable
```

```
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#controller SONET 4/2/0
Router(config-controller)# no ais-shut
Router(config-controller)# framing sonet
Router(config-controller)# clock source line
Router(config-controller)# sts-1 1
Router(config-ctrlr-sts1)# mode vt-15
Router(config-ctrlr-sts1)# vtg 1 t1 1 channel-group 0 timeslots 1-24
Router(config-ctrlr-sts1)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Multilink1
Router(config-if)# ip address 10.1.1.4 255.255.255.0
Router(config-if)# carrier-delay msec 1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# ppp multilink endpoint string mlp_aps_1
Router(config-if)# ppp timeout retry 0 250
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Serial4/2/0.1/1/1:0
Router(config-if)# no ip address
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# controller sonet 3/2/0
Router(config-controller)# shutdown
Router(config-controller)# aps group 1
Router(config-controller)# aps working 1
Router(config-controller)# aps interchassis group 1
Router(config-controller)# no shutdown
Router(config-controller)# end
```

Example Configuring Stateful MLPPP with MR-APS on a Protect Router

This example shows how to configure Stateful MLPPP with MR-APS on a Protect router.

```
Protect-Router> enable
Protect-Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# interchassis group 1
Router(config-r-ic)# monitor peer bfd
Router(config-r-ic)# member ip 10.1.1.7
Router(config-r-ic)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface GigabitEthernet2/1/0
Router(config-if)# ip address 10.1.1.8 255.255.255.0
Router(config-if)# load-interval 30
Router(config-if)# negotiation auto
Router(config-if)# mpls ip
Router(config-if)# mpls label protocol both
Router(config-if)# bfd interval 50 min_rx 150 multiplier 3
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config-if)# interface GigabitEthernet2/1/1
Router(config-if)# ip address 10.1.1.9 255.255.255.0
Router(config-if)# negotiation auto
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)#controller SONET 3/2/0
Router(config-controller)# no ais-shut
Router(config-controller)# framing sonet
Router(config-controller)# clock source line
Router(config-controller)# sts-1 1
Router(config-ctrlr-sts1)# mode vt-15
Router(config-ctrlr-sts1)# vtg 1 t1 1 channel-group 0 timeslots 1-24
Router(config-ctrlr-sts1)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Multilink1
Router(config-if)# ip address 10.1.1.10 255.255.255.0
Router(config-if)# carrier-delay msec 1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# ppp multilink endpoint string mlp_aps_1
Router(config-if)# ppp timeout retry 0 250
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Serial3/2/0.1/1/1:0
Router(config-if)# no ip address
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# end
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# controller sonet 3/2/0
Router(config-controller)# shut
Router(config-controller)# aps group 1
Router(config-controller)# aps protect 1 10.1.1.3
Router(config-controller)# aps interchassis group 1
Router(config-controller)# no shutdown
Router(config-controller)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
WAN commands: complete command syntax, command mode, defaults, usage guidelines and examples	Wide-Area Networking Command Reference
Layer 2 Tunnel Protocol Version 3	*Layer 2 Tunneling Protocol Version 3*
Any Transport over MPLS	*Any Transport over MPLS*
Cisco 12000 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.0S*
Cisco 7600 series routers hardware support	*Cross-Platform Release Notes for Cisco IOS Release 12.2SR*
Cisco 3270 series routers hardware support	*Release Notes for Cisco IOS Software Release 12.2SE*

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-pwe3-frame-relay-03.txt.	*Encapsulation Methods for Transport of Frame Relay over MPLS Networks*
draft-martini-l2circuit-encap-mpls-04.txt.	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-ietf-pwe3-ethernet-encap-08.txt.	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	*Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks*
draft-ietf-ppvpn-l2vpn-00.txt.	*An Architecture for L2VPNs*

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Stateful MLPPP with MR-APS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 18: Feature Information for the Stateful MLPPP with MR-APS feature

Feature Name	Releases	Feature Information
Stateful MLPPP with MR-APS	15.1(3)S	The Stateful MLPPP with MR-APS feature supports IC-SSO for MLPPP sessions, thereby allowing MR-APS from one router to another, while maintaining the MLPPP sessions and avoiding session renegotiation. This feature is available only on Cisco 7600 series routers. In Cisco IOS Release 15.1(3)S, this feature was introduced on the Cisco 7600 series routers. The following commands were introduced or modified: **aps interchassis group**, **debug rgf detail**, **debug rgf error**, **debug rgf event**, **show ccm group all**, **show ccm group id**, **show ccm session id**, **show rgf groups**, **show rgf history**, **show rgf statistics**.