



## **Wide-Area Networking Configuration Guide: Layer 2 Services, Cisco IOS XE Everest 16.6 (Enterprise Routing)**

### **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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2018 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### CHAPTER 1

#### Read Me First 1

---

### CHAPTER 2

#### Wide-Area Networking Overview 3

Finding Feature Information 3

Frame Relay 3

Frame Relay-ATM Internetworking 6

Layer 2 Virtual Private Network 6

Layer 2 Tunneling Protocol Version 3 6

L2VPN Pseudowire Redundancy 6

Layer 2 Virtual Private Network Interworking 7

Layer 2 Local Switching 7

---

### CHAPTER 3

#### Layer 2 Tunneling Protocol Version 3 9

Finding Feature Information 9

Prerequisites for Layer 2 Tunneling Protocol Version 3 9

Restrictions for Layer 2 Tunneling Protocol Version 3 10

General L2TPv3 Restrictions 10

VLAN-Specific Restrictions 11

IPv6 Protocol Demultiplexing for L2TPv3 Restrictions 11

L2TPv3 Control Message Hashing Restrictions 11

L2TPv3 Digest Secret Graceful Switchover Restrictions 11

Quality of Service Restrictions in L2TPv3 Tunneling 12

Information About Layer 2 Tunneling Protocol Version 3 14

L2TPv3 Header Description 14

Session ID 14

Session Cookie 14

Pseudowire Control Encapsulation	15
L2TPv3 Operation	15
L2TPv3 Features	16
Static L2TPv3 Sessions	16
Dynamic L2TPv3 Sessions	17
Control Channel Parameters	17
L2TPv3 Control Channel Authentication Parameters	17
Ethernet over L2TPv3	18
GEC over L2TPv3	19
Sequencing	20
L2TPv3 Type of Service Marking	20
Keepalive	20
MTU Handling	21
L2TPv3 Control Message Hashing	21
L2TPv3 Control Message Rate Limiting	22
L2TPv3 Digest Secret Graceful Switchover	22
L2TPv3 Pseudowire	23
Manual Clearing of L2TPv3 Tunnels	23
L2TPv3 Tunnel Management	23
L2TPv3 Protocol Demultiplexing	24
L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations	24
HDLC over L2TPv3	24
L2TPv3 Benefits	24
Supported L2TPv3 Payloads	25
Ethernet	25
VLAN	25
IPv6 Protocol Demultiplexing	26
Performance Impact of L2TPv3 on Cisco ASR 1000 Series Routers	27
How to Configure Layer 2 Tunneling Protocol Version 3	27
Configuring L2TP Control Channel Parameters	27
Configuring L2TP Control Channel Timing Parameters	28
Configuring L2TPv3 Control Channel Authentication Parameters	29
Configuring L2TP Control Channel Maintenance Parameters	35
Configuring the L2TPv3 Pseudowire	36

Configuring the Xconnect Attachment Circuit	39
Manually Configuring L2TPv3 Session Parameters	41
Configuring Protocol Demultiplexing for L2TPv3	43
Configuring Protocol Demultiplexing for Ethernet Interfaces	43
Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations	44
Configuring GEC over L2TPv3	45
Configuring GEC with Dot1Q	47
Configuring GEC with QinQ	48
Manually Clearing L2TPv3 Tunnels	50
Configuration Examples for Layer 2 Tunneling Protocol Version 3	50
Example: Configuring a Static L2TPv3 Session for an Xconnect Ethernet Interface	51
Example: Configuring a Negotiated L2TPv3 Session for an Xconnect VLAN Subinterface	51
Example: Configuring a Negotiated L2TPv3 Session for Local HDLC Switching	51
Example: Verifying an L2TPv3 Session	52
Example: Verifying an L2TP Control Channel	53
Example: Configuring L2TPv3 Control Channel Authentication	53
Example: Configuring L2TPv3 Digest Secret Graceful Switchover	54
Example: Verifying L2TPv3 Digest Secret Graceful Switchover	54
Example: Configuring a Pseudowire Class for Fragmentation of IP Packets	55
Example: Configuring Protocol Demultiplexing for L2TPv3	55
Example: Manually Clearing an L2TPv3 Tunnel	55
Example: Configuring an L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations	55
Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport	55
Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Dynamic Mode	55
Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Static Mode	56
Example: Configuring GEC over L2TPv3	56
Example: Configuring GEC with Dot1q over L2TPv3	56
Example: Configuring GEC with QinQ over L2TPv3	57
Additional References	57
Feature Information for Layer 2 Tunneling Protocol Version 3	58
Glossary	59

Finding Feature Information	61
Prerequisites for L2VPN Pseudowire Redundancy	61
Restrictions for L2VPN Pseudowire Redundancy	62
Information About L2VPN Pseudowire Redundancy	62
Introduction to L2VPN Pseudowire Redundancy	62
How to Configure L2VPN Pseudowire Redundancy	64
Configuring the Pseudowire	64
Configuring the Pseudowire using the commands associated with the L2VPN Protocol-Based CLIs feature	65
Configuring L2VPN Pseudowire Redundancy	66
Configuring L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature	68
Forcing a Manual Switchover to the Backup Pseudowire VC	70
Verifying the L2VPN Pseudowire Redundancy Configuration	71
Verifying the L2VPN Pseudowire Redundancy Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature	72
Configuration Examples for L2VPN Pseudowire Redundancy	74
Example L2VPN Pseudowire Redundancy and AToM (Like to Like)	75
Example L2VPN Pseudowire Redundancy and L2VPN Interworking	75
Example L2VPN Pseudowire Redundancy with Layer 2 Local Switching	76
Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3	76
Configuration Examples for L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature	77
Example L2VPN Pseudowire Redundancy and AToM (Like to Like) using the commands associated with the L2VPN Protocol-Based CLIs feature	77
Example L2VPN Pseudowire Redundancy and L2VPN Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature	78
Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3 using the commands associated with the L2VPN Protocol-Based CLIs feature	79
Additional References	81
Feature Information for L2VPN Pseudowire Redundancy	82

**CHAPTER 5****Layer 2 Local Switching 83**

Finding Feature Information	83
Prerequisites for Layer 2 Local Switching	84

Restrictions for Layer 2 Local Switching	84
Information About Layer 2 Local Switching	84
Layer 2 Local Switching Overview	84
NSF SSO—Local Switching Overview	84
Layer 2 Local Switching Applications	84
How to Configure Layer 2 Local Switching	85
Configuring Ethernet VLAN Same-Port Switching	85
Configuring Ethernet Port Mode to Ethernet VLAN Local Switching	87
Configuring ATM-to-ATM PVC Local Switching and Same-Port Switching	88
Configuring ATM-to-ATM PVP Local Switching	89
Configuring ATM PVP Same-Port Switching	90
Configuring Frame Relay-to-Frame Relay Local Switching	91
Verifying Layer 2 Local Switching	93
Verifying Layer 2 Local Switching Configuration	93
Verifying the NSF SSO Local Switching Configuration	93
Troubleshooting Tips	94
Configuration Examples for Layer 2 Local Switching	94
Example: Configuring Ethernet VLAN Same-Port Switching	94
Example: Configuring NSF SSO Ethernet Port Mode to Ethernet VLAN Local Switching	95
Example: Configuring ATM-to-ATM Local Switching	97
Example: Configuring ATM PVC Same-Port Switching	97
Example: Configuring ATM PVP Same-Port Switching	97
Example: Configuring Frame Relay-to-Frame Relay Local Switching	97
Additional References	98
Feature Information for Layer 2 Local Switching	99







# CHAPTER 1

## Read Me First

---

### Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

### Feature Information

Use [Cisco Feature Navigator](#) to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

### Related References

- [Cisco IOS Command References, All Releases](#)

### Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the . RSS feeds are a free service.





## CHAPTER 2

# 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, on page 3](#)
- [Frame Relay, on page 3](#)
- [Layer 2 Virtual Private Network, on page 6](#)

## 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](http://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 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.

## 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 I2tpext 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.







## CHAPTER 3

# 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 I2tpext 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, on page 9](#)
- [Prerequisites for Layer 2 Tunneling Protocol Version 3, on page 9](#)
- [Restrictions for Layer 2 Tunneling Protocol Version 3, on page 10](#)
- [Information About Layer 2 Tunneling Protocol Version 3, on page 14](#)
- [How to Configure Layer 2 Tunneling Protocol Version 3, on page 27](#)
- [Configuration Examples for Layer 2 Tunneling Protocol Version 3, on page 50](#)
- [Additional References, on page 57](#)
- [Feature Information for Layer 2 Tunneling Protocol Version 3, on page 58](#)
- [Glossary, on page 59](#)

## 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](http://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.
- Only Ethernet, HDLC, Frame Relay and VLAN (802.1Q, QinQ, and QinAny) attachment circuits are supported; EVC is not supported.
- The IP local interface must be a loopback interface and the loopback interface cannot be in a VRF. Configuring any other interface with the "ip local interface" command results in a nonoperational setting.

## 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.

## IPv6 Protocol Demultiplexing for L2TPv3 Restrictions

- IPv6 protocol demultiplexing is supported only for Ethernet traffic.
- IPv6 protocol demultiplexing is supported over noninterworking sessions.

## 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 task: "Configuring QoS for L2TPv3 on the Cisco 7500 Series Example".)

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 L2TP Control Channel Parameters](#).

### 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 restriction that 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**

#### 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.

#### 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.

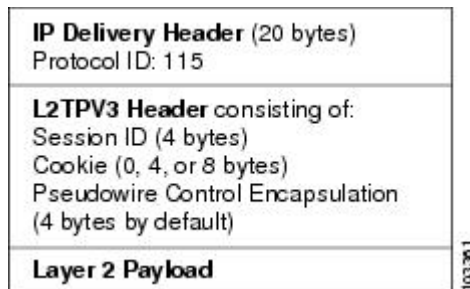
## Information About Layer 2 Tunneling Protocol Version 3

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.

### 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 1: 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 [Configuring L2TP Control Channel Parameters](#) section for more information on the CLI commands for L2TPv3.

### Session ID

The L2TPv3 session ID 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. 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 configured manually for static sessions or determined dynamically 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 (see the [Sequencing](#) section). 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 Operation

L2TPv3 includes the following features:

- Xconnect for Layer 2 tunneling through a pseudowire over an IP network
- Layer 2 VPNs for PE-to-PE device service using xconnect that supports Ethernet and VLAN, including both static and dynamic (using the new L2TPv3 signaling) forwarded sessions

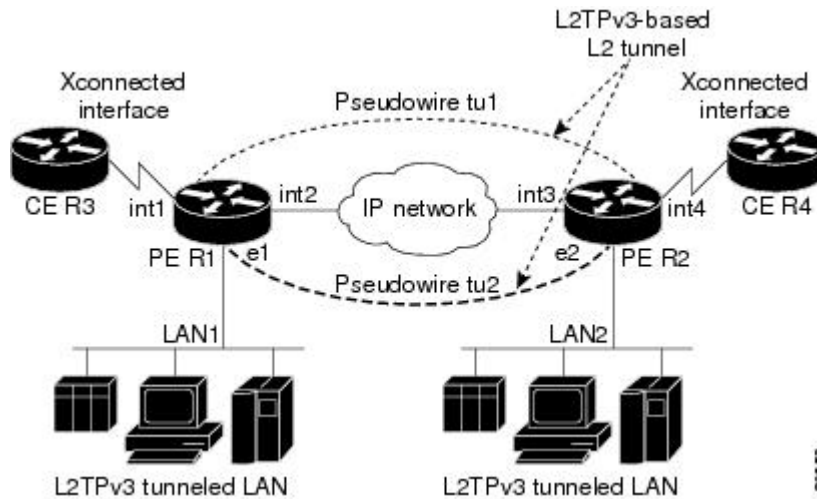
The initial Cisco IOS software supported only the following features:

- 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 the UDP
- Point-to-point sessions, not point-to-multipoint or multipoint-to-point sessions
- Sessions between the same Layer 2 protocols, such as Ethernet-to-Ethernet and VLAN-to-VLAN, but not VLAN-to-Ethernet

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 devices of the IP network treat the traffic as any other IP traffic and need not know anything about the customer networks.

Figure 2: L2TPv3 Operation



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

In this example, the customer edge (CE) devices 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 the 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 is 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 Features

L2TPv3 provides xconnect support for Ethernet and VLAN using Static and Dynamic sessions.

### 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.

## 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 1: Compatibility Matrix for L2TPv3 Authentication Methods**

<b>PE1 Authentication Configuration</b>	<b>PE2 Supporting Old Authentication<sup>1</sup></b>	<b>PE2 Supporting New Authentication<sup>2</sup></b>	<b>PE2 Supporting Old and New Authentication<sup>3</sup></b>
None	None	None New integrity check	None New integrity check
Old authentication	Old authentication	—	Old authentication <b>Old authentication</b> and new authentication <b>Old authentication</b> and new integrity check
New authentication	—	New authentication	New authentication Old authentication and <b>new authentication</b>
New integrity check	None	None New integrity check	None New integrity check
Old and new authentication	Old authentication	New authentication	Old authentication New authentication <b>Old and new authentication</b> <b>Old authentication</b> and new integrity check
Old authentication and new integrity check	Old authentication	—	Old authentication <b>Old authentication</b> and new authentication <b>Old authentication</b> and new integrity check

<sup>1</sup> Any PE software that supports only the old CHAP-like authentication system.

<sup>2</sup> 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.

<sup>3</sup> Any PE software that supports both the old CHAP-like authentication and the new message digest authentication and integrity checking authentication system.

## Ethernet over L2TPv3

The Ethernet over L2TPv3 feature provides support for Ethernet-based Layer 2 payload tunneling over IP core networks using L2TPv3.

The Ethernet over L2TPv3 feature supports the following like-to-like switching modes:

- Ethernet port mode
- Ethernet VLAN mode
- Ethernet VLAN mode with VLAN rewrite
- Ethernet QinQ and QinAny mode



---

**Note** The QinQ over L2TPv3 support feature includes QinAny over L2TPv3, which has a fixed outer VLAN tag and a variable inner VLAN tag.

---

The Ethernet over L2TPv3 feature supports the following types of interworking:

- Ethernet port to VLAN (routed)
- Ethernet port to VLAN (bridged)
- QinQ to Ethernet VLAN or Port Interworking (routed)
- QinQ to Ethernet VLAN or Port Interworking (bridged)



---

**Note** QinAny Interworking is not a valid configuration because the inner VLAN tag is undetermined.

---

## GEC over L2TPv3

Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3) provides support for GEC-based Layer 2 payload tunneling over IP core networks using L2TPv3. GEC also known as *port channel* is integrated with Ethernet and dot1q attachment circuits (ACs).

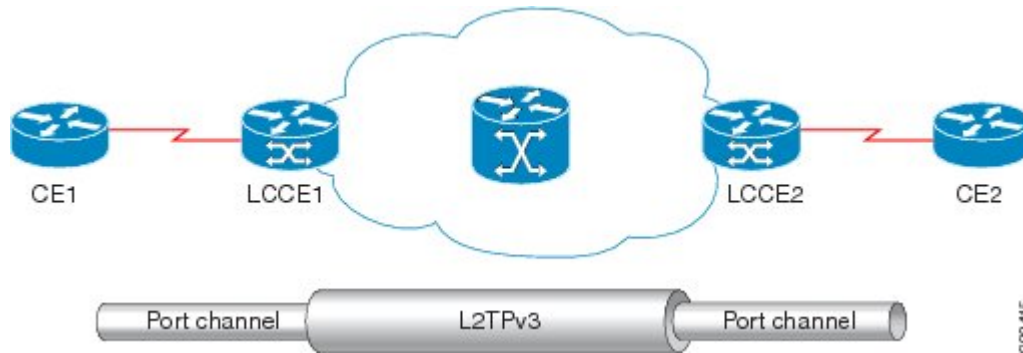
A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

Interworking switching is supported in the following scenarios:

- The customer-edge-provider-edge (CE-PE) connecting interface on the local PE is a port-channel interface without dot1q encapsulation. The CE-PE connecting interface on the remote PE is a port-channel interface with dot1q encapsulation.
- The CE-PE connecting interface on the local PE is a port-channel interface with or without dot1q encapsulation. The CE-PE connecting interface on the remote PE is an Ethernet interface with or without dot1q encapsulation.

The figure below illustrates a port channel over IP core networks using L2TPv3. CE1 and CE2 are connected to L2TP Control Connection Endpoints (LCCE) and through port channels. The LCCE is connected to the IP core network using L2TPv3.

Figure 3: GEC over L2TPv3



## 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 I2tpext 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.

## 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](#).

## 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](#).

## 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.

## HDLC over L2TPv3

HDLC for Layer 2 Data Encapsulation provides encapsulation of port-to-port Layer 2 traffic. All HDLC traffic including IPv4, IPv6, and non-IP packet, such as IS-IS, is tunneled over L2TPv3. HDLC does not support interworking mode.




---

**Note** L2TPv3 supports the IPv4 tunnel only for HDLC. The IPv4 tunnel supports IPv4 and IPv6 packets.

---

## 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)

## 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.

---

### 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.



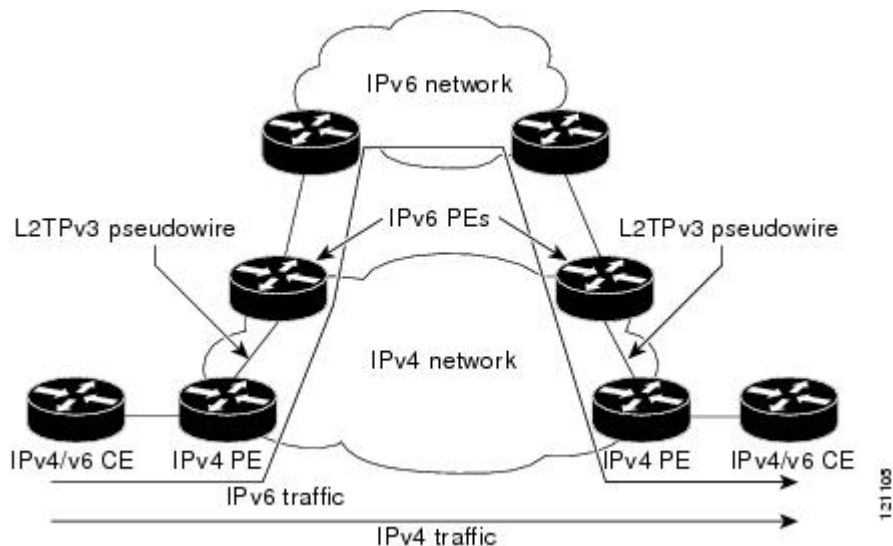
**Note** Because of the way in which L2TPv3 handles VLAN packets, the Ethernet interface must be configured in promiscuous mode to capture all traffic received on the Ethernet segment attached to the device. All frames are tunneled through the L2TP pseudowire.

## 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 4: 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 2: Valid Configuration Scenarios

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

## Performance Impact of L2TPv3 on Cisco ASR 1000 Series Routers

L2TPv3 supports the following maximum number of attachment circuits and tunnels:

- First-generation Cisco ASR 1000 Series Route Processor (RP1) with Embedded Services Processor 10 (ESP10)
  - Attachment circuits for Ethernet: 8000 per system in a typical user environment. This includes 4000 per port and 8000 per SPA.
  - L2TPv3 tunnels: 1000 (in a typical user environment) and 2000 (maximum).
- Second-generation Cisco ASR 1000 Series Route Processor (RP2) with Embedded Services Processor 20 (ESP20)
  - Attachment circuits for Ethernet: 16,000 per system in a typical user environment. This includes 4000 per port and 8000 per SPA.
  - L2TPv3 tunnels: 2000 (in a typical user environment) and 4000 (maximum).

L2TPv3 adds tunnel encapsulation to TCP packets, which can cause fragmentation of big packets (packet size larger than the session MTU). Consider a scenario where a big TCP packet is followed by a small TCP packet (packet size smaller than the session MTU). After L2TPv3 encapsulation, the encapsulated big TCP packet will be fragmented, but the encapsulated small TCP packet will not be fragmented. On the Cisco ASR 1000 Series Routers, the fragmentation and reassembly of the big TCP packet requires an additional processor cycle. Because Cisco ASR 1000 Series Routers follow multithread processing, the small packet will need shorter processing time and may be forwarded ahead of the fragmented big packet. This process may result in packet sequence changes on the receiver's end.

As a workaround, you can enable the `ip pmtu` command to prevent the fragmentation of tunneled packets (see the "[MTU Handling](#)" section).

## How to Configure Layer 2 Tunneling Protocol Version 3

### 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, default values are applied.

### SUMMARY STEPS

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

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>l2tp-class</b> [ <i>l2tp-class-name</i> ]  <b>Example:</b> Device(config)# l2tp-class class1	Specifies the L2TP class name and enters L2TP class configuration mode.  • The <i>l2tp-class-name</i> argument is optional. However, to configure multiple L2TP classes, you must specify a unique <i>l2tp-class-name</i> for each one.
<b>Step 4</b>	<b>retransmit</b> { <b>initial retries</b> <i>initial-retries</i>   <b>retries</b> <i>retries</i>   <b>timeout</b> { <b>max</b>   <b>min</b> } <i>timeout</i> }	(Optional) Configures parameters that affect the retransmission of control packets.
	<b>Example:</b> Device(config-l2tp-class)# retransmit retries 10	<ul style="list-style-type: none"> <li>• <b>initial retries</b>—Specifies how many SCCRQs are resent before giving up on the session. Valid values for the <i>initial-retries</i> argument range from 1 to 1000. The default value is 2.</li> <li>• <b>retries</b>—Specifies how many retransmission cycles occur before determining that the peer PE device is</li> </ul>

	Command or Action	Purpose
		<p>not responding. Valid values for the <i>retries</i> argument range from 1 to 1000. The default value is 15.</p> <ul style="list-style-type: none"> <li>• <b>timeout</b> {<i>max</i>   <i>min</i>}—Specifies maximum and minimum retransmission intervals (in seconds) for resending control packets. Valid values for the <i>timeout</i> argument range from 1 to 8. The default maximum interval is 8. The default minimum interval is 1.</li> </ul>
<b>Step 5</b>	<p><b>timeout setup</b> <i>seconds</i></p> <p><b>Example:</b></p> <pre>Device(config-l2tp-class)# timeout setup 400</pre>	<p>(Optional) Configures the amount of time, in seconds, allowed to set up a control channel.</p> <ul style="list-style-type: none"> <li>• Valid values for the <i>seconds</i> argument range from 60 to 6000. The default value is 300.</li> </ul>
<b>Step 6</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>Device(config-l2tp-class)# exit</pre>	Exits L2TP class configuration mode.

## 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
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>l2tp-class</b> [ <i>l2tp-class-name</i> ] <b>Example:</b> Device(config)# l2tp-class class1	Specifies the L2TP class name and enters L2TP class configuration mode. <ul style="list-style-type: none"> <li>• The <i>l2tp-class-name</i> argument is optional. However, to configure multiple L2TP classes, you must specify a unique <i>l2tp-class-name</i> for each one.</li> </ul>
<b>Step 4</b>	<b>authentication</b> <b>Example:</b> Device(config-l2tp-class)# authentication	(Optional) Enables authentication for the control channel between PE devices.
<b>Step 5</b>	<b>password</b> [ <b>0</b>   <b>7</b> ] <i>password</i> <b>Example:</b> Device(config-l2tp-class)# password cisco	(Optional) Configures the password used for control channel authentication. <ul style="list-style-type: none"> <li>• [<b>0</b>   <b>7</b>]—(Optional) Specifies the input format of the shared secret. The default value is <b>0</b>. <ul style="list-style-type: none"> <li>• <b>0</b>—Specifies that a plain-text secret is entered.</li> <li>• <b>7</b>—Specifies that an encrypted secret is entered.</li> </ul> </li> <li>• <i>password</i>—Defines the shared password between peer devices.</li> </ul>
<b>Step 6</b>	<b>hostname</b> <i>name</i> <b>Example:</b> Device(config-l2tp-class)# hostname yb2	(Optional) Specifies a hostname used to identify the device during L2TP control channel authentication. <ul style="list-style-type: none"> <li>• If you do not use this command, the default hostname of the device is used.</li> </ul>
<b>Step 7</b>	<b>exit</b> <b>Example:</b> 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
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>l2tp-class</b> [ <i>l2tp-class-name</i> ] <b>Example:</b> Device(config)# l2tp-class class1	Specifies the L2TP class name and enters L2TP class configuration mode. <ul style="list-style-type: none"> <li>• The <i>l2tp-class-name</i> argument is optional. However, to configure multiple L2TP classes, you must specify a unique <i>l2tp-class-name</i> for each one.</li> </ul>
<b>Step 4</b>	<b>digest</b> [ <b>secret</b> [ <b>0</b>   <b>7</b> ] <i>password</i> ] [ <b>hash</b> { <b>md5</b>   <b>sha</b> }] <b>Example:</b> Device(config-l2tp-class)# digest secret cisco hash sha	(Optional) Enables L2TPv3 control channel authentication or integrity checking. <ul style="list-style-type: none"> <li>• <b>secret</b>—(Optional) Enables L2TPv3 control channel authentication.</li> </ul> <p><b>Note</b> If the <b>digest</b> command is issued without the <b>secret</b> keyword option, L2TPv3 integrity checking is enabled.</p> <ul style="list-style-type: none"> <li>• [<b>0</b>   <b>7</b>]—Specifies the input format of the shared secret. The default value is <b>0</b>.               <ul style="list-style-type: none"> <li>• <b>0</b>—Specifies that a plain-text secret is entered.</li> <li>• <b>7</b>—Specifies that an encrypted secret is entered.</li> </ul> </li> <li>• <i>password</i>—Defines the shared secret between peer devices. The value entered for the <i>password</i> argument must be in the format that matches the input format specified by the [<b>0</b>   <b>7</b>] keyword option.</li> <li>• <b>hash</b> {<b>md5</b>   <b>sha</b>}—(Optional) Specifies the hash function to be used in per-message digest calculations.               <ul style="list-style-type: none"> <li>• <b>md5</b>—Specifies HMAC-MD5 hashing.</li> <li>• <b>sha</b>—Specifies HMAC-SHA-1 hashing.</li> </ul> </li> </ul>

	Command or Action	Purpose
		The default hash function is <b>md5</b> .
<b>Step 5</b>	<b>digest check</b> <b>Example:</b> <pre>Device(config-l2tp-class)# digest check</pre>	(Optional) Enables the validation of the message digest in received control messages. <ul style="list-style-type: none"> <li>• Validation of the message digest is enabled by default.</li> </ul> <b>Note</b> Validation of the message digest cannot be disabled if authentication has been enabled using the <b>digest secret</b> command. If authentication has not been configured with the <b>digest secret</b> command, the digest check can be disabled to increase performance.
<b>Step 6</b>	<b>hidden</b> <b>Example:</b> <pre>Device(config-l2tp-class)# hidden</pre>	(Optional) Enables AV pair hiding when sending control messages to an L2TPv3 peer. <ul style="list-style-type: none"> <li>• AV pair hiding is disabled by default.</li> <li>• Only the hiding of the cookie AV pair is supported.</li> <li>• If a cookie is configured in L2TP class configuration mode (see the section "<i>Manually Configuring L2TPv3 Session Parameters</i>"), enabling AV pair hiding causes that cookie to be sent to the peer as a hidden AV pair using the password configured with the <b>digest secret</b> command.</li> </ul> <b>Note</b> AV pair hiding is enabled only if authentication has been enabled using the <b>digest secret</b> command, and no other authentication method is configured.
<b>Step 7</b>	<b>exit</b> <b>Example:</b> <pre>Device(config-l2tp-class)# exit</pre>	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	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>l2tp-class</b> <i>l2tp-class-name</i> <b>Example:</b> Device(config)# l2tp-class class1	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 4	<b>digest</b> [secret [0   7] <i>password</i> ] [hash {md5   sha}] <b>Example:</b> Device(config-l2tp-class)# digest secret cisco2 hash sha	Configures a new password to be used in L2TPv3 control channel authentication. <ul style="list-style-type: none"><li>• A maximum of two passwords may be configured at any time.</li></ul> <b>Note</b> Authentication will now occur using both the old and new passwords.
Step 5	<b>end</b> <b>Example:</b> Device(config-l2tp-class)# end	Ends your configuration session by exiting to privileged EXEC mode.
Step 6	<b>show l2tun tunnel all</b> <b>Example:</b> 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. <ul style="list-style-type: none"><li>• Tunnels should be updated with the new control channel authentication password within a matter of</li></ul>

	Command or Action	Purpose
		<p>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 <a href="#">“Manually Clearing L2TPv3 Tunnels.”</a>”</p> <p><b>Note</b> 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.</p>
<b>Step 7</b>	<p><b>configure terminal</b></p> <p><b>Example:</b></p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
<b>Step 8</b>	<p><b>l2tp-class</b> [<i>l2tp-class-name</i>]</p> <p><b>Example:</b></p> <pre>Device(config)# l2tp-class class1</pre>	<p>Specifies the L2TP class name and enters L2TP class configuration mode.</p> <ul style="list-style-type: none"> <li>The <i>l2tp-class-name</i> argument is optional. However, to configure multiple L2TP classes, you must specify a unique <i>l2tp-class-name</i> for each one.</li> </ul>
<b>Step 9</b>	<p><b>no digest</b> [<b>secret</b> [0   7] <i>password</i> [<b>hash</b> {md5   sha}]]</p> <p><b>Example:</b></p> <pre>Device(config-l2tp-class)# no digest secret cisco hash sha</pre>	<p>Removes the old password used in L2TPv3 control channel authentication.</p> <p><b>Note</b> Do not remove the old password until all peer PE devices have been updated with the new password.</p>
<b>Step 10</b>	<p><b>end</b></p> <p><b>Example:</b></p> <pre>Device(config-l2tp-class)# end</pre>	Ends your configuration session by exiting to privileged EXEC mode.
<b>Step 11</b>	<p><b>show l2tun tunnel all</b></p> <p><b>Example:</b></p> <pre>Device# show l2tun tunnel all</pre>	<p>(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.</p> <ul style="list-style-type: none"> <li>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 manually and a defect report should be filed with TAC. To clear an L2TPv3 tunnel manually, perform the task described in the section <a href="#">“Manually Clearing L2TPv3 Tunnels.”</a>”</li> </ul>

	Command or Action	Purpose
		<p><b>Note</b> 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.</p>

## 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
<b>Step 1</b>	<p><b>enable</b></p> <p><b>Example:</b> Device&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<p><b>configure terminal</b></p> <p><b>Example:</b> Device# configure terminal</p>	<p>Enters global configuration mode.</p>
<b>Step 3</b>	<p><b>l2tp-class</b> [<i>l2tp-class-name</i>]</p> <p><b>Example:</b> Device(config)# l2tp-class class1</p>	<p>Specifies the L2TP class name and enters L2TP class configuration mode.</p> <ul style="list-style-type: none"> <li>• The <i>l2tp-class-name</i> argument is optional. However, to configure multiple L2TP classes, you must specify a unique <i>l2tp-class-name</i> for each one.</li> </ul>
<b>Step 4</b>	<p><b>hello</b> <i>interval</i></p> <p><b>Example:</b> Device(config-l2tp-class)# hello 100</p>	<p>(Optional) Specifies the exchange interval (in seconds) used between L2TP hello packets.</p> <ul style="list-style-type: none"> <li>• Valid values for the <i>interval</i> argument range from 0 to 1000. The default value is 60.</li> </ul>

	Command or Action	Purpose
<b>Step 5</b>	<b>exit</b>  <b>Example:</b> 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
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>pseudowire-class</b> [ <i>pw-class-name</i> ]  <b>Example:</b> Device(config)# pseudowire-class etherpw	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
<b>Step 4</b>	<b>encapsulation l2tpv3</b>  <b>Example:</b> Device(config-pw)# encapsulation l2tpv3	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.

	Command or Action	Purpose
<b>Step 5</b>	<p><b>protocol</b> {<b>l2tpv3</b>   <b>none</b>} [<i>l2tp-class-name</i>]</p> <p><b>Example:</b></p> <pre>Device(config-pw)# protocol l2tpv3 class1</pre>	<p>(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 "<a href="#">Configuring L2TP Control Channel Parameters</a>").</p> <ul style="list-style-type: none"> <li>If the <i>l2tp-class-name</i> argument is not specified, the default values for L2TP control channel parameters are used. The default <b>protocol</b> option is <b>l2tpv3</b>.</li> <li>If you do not want to use signaling in the L2TPv3 sessions created with this pseudowire class, enter <b>protocol none</b>.</li> </ul>
<b>Step 6</b>	<p><b>ip local interface</b> <i>interface-name</i></p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip local interface e0/0</pre>	<p>Specifies the PE device interface whose IP address is to be used as the source IP address for sending tunneled packets.</p> <ul style="list-style-type: none"> <li>The same or a different local interface name can be used for each of the pseudowire classes configured between a pair of PE devices.</li> </ul> <p><b>Note</b> This command must be configured for pseudowire-class configurations using L2TPv3 as the data encapsulation method.</p>
<b>Step 7</b>	<p><b>ip pmtu</b></p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip pmtu</pre>	<p>(Optional) Enables the discovery of the PMTU for tunneled traffic and helps fragmentation.</p> <ul style="list-style-type: none"> <li>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.</li> </ul> <p><b>Note</b> The <b>ip pmtu</b> command is not supported if you disabled signaling with the <b>protocol none</b> command in Step 5.</p> <ul style="list-style-type: none"> <li>This command must be enabled in the pseudowire class configuration to enable fragmentation of IP packets before the data enters the pseudowire.</li> </ul> <p><b>Note</b> To enable fragmentation of IP packets before the data enters the pseudowire, Cisco recommends that you also enter the <b>ip dfbit set</b> command in pseudowire class configuration mode. This allows the PMTU to be obtained more rapidly.</p>

	Command or Action	Purpose
		<p><b>Note</b> When the <b>ip pmtu</b> 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.</p>
<b>Step 8</b>	<p><b>ip tos</b> {<i>value value</i>   <b>reflect</b>}</p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip tos reflect</pre>	<p>(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.</p> <ul style="list-style-type: none"> <li>Valid values for the <i>value</i> argument range from 0 to 255. The default ToS byte value is 0.</li> </ul>
<b>Step 9</b>	<p><b>ip dfbit set</b></p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip dfbit set</pre>	<p>(Optional) Configures the value of the DF bit in the outer headers of tunneled packets.</p> <ul style="list-style-type: none"> <li>Use this command if (for performance reasons) you do not want reassembly of tunneled packets on the peer PE device.</li> <li>This command is disabled by default.</li> </ul>
<b>Step 10</b>	<p><b>ip ttl</b> <i>value</i></p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip ttl 100</pre>	<p>(Optional) Configures the value of the time to live (TTL) byte in the IP headers of tunneled packets.</p> <ul style="list-style-type: none"> <li>Valid values for the <i>value</i> argument range from 1 to 255. The default TTL byte value is 255.</li> </ul>
<b>Step 11</b>	<p><b>ip protocol</b> {<b>l2tp</b>   <i>protocol-number</i>}</p> <p><b>Example:</b></p> <pre>Device(config-pw)# ip protocol l2tp</pre>	<p>(Optional) Configures the IP protocol to be used for tunneling packets.</p>
<b>Step 12</b>	<p><b>sequencing</b> {<b>transmit</b>   <b>receive</b>   <b>both</b>}</p> <p><b>Example:</b></p> <pre>Device(config-pw)# sequencing both</pre>	<p>(Optional) Specifies the direction in which sequencing of data packets in a pseudowire is enabled:</p> <ul style="list-style-type: none"> <li><b>transmit</b>—Updates the Sequence Number field in the headers of data packets sent over the pseudowire according to the data encapsulation method that is used.</li> <li><b>receive</b>—Keeps the Sequence Number field in the headers of data packets received over the pseudowire. Out-of-order packets are dropped.</li> <li><b>both</b>—Enables both the <b>transmit</b> and <b>receive</b> options.</li> </ul>
<b>Step 13</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>Device(config-pw)# exit</pre>	<p>Exits pseudowire class configuration mode.</p>

## 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	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>interface</b> <i>type slot / port</i> <b>Example:</b> 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	<b>xconnect</b> <i>peer-ip-address vcid pseudowire-parameters</i> [ <b>sequencing</b> { <b>transmit</b>   <b>receive</b>   <b>both</b> }] <b>Example:</b> 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. <ul style="list-style-type: none"> <li>• The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device.</li> <li>• At least one of the following pseudowire class parameters must be configured for the <i>pseudowire-parameters</i> argument:               <ul style="list-style-type: none"> <li>• <b>encapsulation</b> {<b>l2tpv3</b> [<b>manual</b>]   <b>mpls</b>}—Specifies the tunneling method used to encapsulate data in the pseudowire:                   <ul style="list-style-type: none"> <li>• <b>l2tpv3</b>—L2TPv3 is the tunneling method to be used.</li> <li>• <b>manual</b>—(Optional) No signaling is to be used in the L2TPv3 control channel. This command places the device in xconnect configuration mode for the manual</li> </ul> </li> </ul> </li> </ul>

	Command or Action	Purpose
		<p>configuration of L2TPv3 parameters for the attachment circuit.</p> <ul style="list-style-type: none"> <li>• <b>mpls</b>—MPLS is the tunneling method to be used.</li> <li>• <b>pw-class</b> {<i>pw-class-name</i>}—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken.</li> <li>• The optional <b>encapsulation</b> parameter specifies the method of pseudowire tunneling used: L2TPv3 or MPLS. Enter <b>manual</b> if you do not want signaling to be used in the L2TPv3 control channel. The <b>encapsulation l2tpv3 manual</b> keyword combination enters xconnect configuration submode. See the section "<i>Manually Configuring L2TPv3 Session Parameters</i>" for the other L2TPv3 commands that you must enter to complete the configuration of the L2TPv3 control channel. If you do not enter an <b>encapsulation</b> value, the encapsulation method entered with the <b>password</b> command in the Configuring the Xconnect Attachment Circuit task is used.</li> <li>• The optional <b>pw-class</b> 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.</li> </ul> <p><b>Note</b> You must configure either the <b>encapsulation</b> or the <b>pw-class</b> option or both.</p> <p><b>Note</b> If you select L2TPv3 as your data encapsulation method, you must specify the <b>pw-class</b> keyword.</p> <ul style="list-style-type: none"> <li>• The optional <b>sequencing</b> parameter specifies whether sequencing is required for packets that are received, sent, or both received and sent.</li> </ul>
<b>Step 5</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>Device(config-if)# exit</pre>	Exits interface configuration mode.



## Manually Configuring L2TPv3 Session Parameters

When you bind an attachment circuit to an L2TPv3 pseudowire for the xconnect service by using the **xconnect l2tpv3 manual** command (see the section "[Configuring the Xconnect Attachment Circuit](#)") because you do not want signaling, you must configure L2TP-specific parameters to complete the L2TPv3 control channel configuration.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **xconnect** *peer-ip-address vc-id encapsulation l2tpv3 manual pw-class pw-class-name*
5. **l2tp id** *local-session-id remote-session-id*
6. **l2tp cookie local** *size low-value [high-value]*
7. **l2tp cookie remote** *size low-value [high-value]*
8. **l2tp hello** *l2tp-class-name*
9. **exit**
10. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>interface</b> <i>type slot / port</i> <b>Example:</b> 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	<b>xconnect</b> <i>peer-ip-address vc-id encapsulation l2tpv3 manual pw-class pw-class-name</i> <b>Example:</b> Device(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual 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, and enters xconnect configuration mode. <ul style="list-style-type: none"><li>• The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device.</li><li>• The <b>encapsulation l2tpv3 manual</b> parameter specifies that L2TPv3 is to be used as the pseudowire tunneling method.</li><li>• The mandatory <b>pw-class</b> <i>pw-class-name</i> keyword and argument combination specifies the pseudowire</li></ul>

	Command or Action	Purpose
		class configuration from which the data encapsulation type (L2TPv3) is taken.
<b>Step 5</b>	<b>l2tp id</b> <i>local-session-id remote-session-id</i> <b>Example:</b> Device(config-if-xconn)# l2tp id 222 111	Configures the identifiers for the local L2TPv3 session and for the remote L2TPv3 session on the peer PE device. <ul style="list-style-type: none"> <li>This command is required to complete the attachment circuit configuration and a static L2TPv3 session configuration.</li> </ul>
<b>Step 6</b>	<b>l2tp cookie local</b> <i>size low-value [high-value]</i> <b>Example:</b> Device(config-if-xconn)# l2tp cookie local 4 54321	(Optional) Specifies the value that the peer PE must include in the cookie field of incoming (received) L2TP packets. <ul style="list-style-type: none"> <li>The size of the cookie field can be 4 or 8 bytes. If you do not enter this command, no cookie value is included in the header of L2TP packets.</li> <li>If you configure the cookie length in incoming packets as 8 bytes, you must specify a 4-byte high value and a 4-byte low value.</li> </ul>
<b>Step 7</b>	<b>l2tp cookie remote</b> <i>size low-value [high-value]</i> <b>Example:</b> Device(config-if-xconn)# l2tp cookie remote 4 12345	(Optional) Specifies the value that the device includes in the cookie field of outgoing (sent) L2TP packets. <ul style="list-style-type: none"> <li>The size of the cookie field can be 4 or 8 bytes. If you do not enter this command, no cookie value is included in the header of L2TP packets.</li> <li>If you configure the cookie length in outgoing packets as 8 bytes, you must specify a 4-byte high value and a 4-byte low value.</li> </ul>
<b>Step 8</b>	<b>l2tp hello</b> <i>l2tp-class-name</i> <b>Example:</b> Device(config-if-xconn)# l2tp hello l2tp-defaults	(Optional) Specifies the L2TP class name to be used (see the section " <a href="#">Configuring L2TP Control Channel Parameters</a> ") for control channel configuration parameters, including the interval to use between hello keepalive messages. <p><b>Note</b> This command assumes that there is no control plane to negotiate control channel parameters and that a control channel is to be used to provide keepalive support through an exchange of L2TP hello messages. By default, no hello messages are sent.</p>
<b>Step 9</b>	<b>exit</b> <b>Example:</b> Device(config-if-xconn)# exit	Exits xconnect configuration mode.

	Command or Action	Purpose
Step 10	<b>exit</b>  <b>Example:</b> Device(config-if)# exit	Exits interface configuration mode.

## 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	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>interface</b> <i>type slot / port</i>  <b>Example:</b> Device(config)# interface ethernet 0/1	Specifies the interface by type, slot, and port number, and enters interface configuration mode.
Step 4	<b>ip address</b> <i>ip-address mask [secondary]</i>  <b>Example:</b> Device(config-if)# ip address 172.16.128.4	Sets a primary or secondary IP address for an interface.
Step 5	<b>xconnect</b> <i>peer-ip-address vcid pw-class pw-class-name</i>  <b>Example:</b>	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.

	Command or Action	Purpose
	<pre>Device(config-if)# xconnect 10.0.3.201 888 pw-class demux</pre>	<ul style="list-style-type: none"> <li>The peer device ID (IP address) and virtual circuit ID must be a unique combination on the device.</li> <li><b>pw-class</b> <i>pw-class-name</i>—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The <b>pw-class</b> 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.</li> </ul> <p><b>Note</b> The L2TPv3 session can also be provisioned manually. See the section "<i>Manually Configuring L2TPv3 Session Parameters</i>" for information about manually configuring the L2TPv3 session parameters.</p>
<b>Step 6</b>	<p><b>match protocol ipv6</b></p> <p><b>Example:</b></p> <pre>Device(config-if-xconn)# match protocol ipv6</pre>	Enables protocol demultiplexing of IPv6 traffic.
<b>Step 7</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>Device(config-if-xconn)# exit</pre>	Exits xconnect configuration mode.
<b>Step 8</b>	<p><b>exit</b></p> <p><b>Example:</b></p> <pre>Device(config-if)# exit</pre>	Exits interface configuration mode.

## 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	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>interface</b> <i>type number</i> <b>Example:</b> Device(config)# interface gigabitethernet 1/0/0	Specifies an interface and enters interface configuration mode.
Step 4	<b>dot1q tunneling ethertype</b> {0x88A8   0x9100   0x9200} <b>Example:</b> 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	<b>exit</b> <b>Example:</b> Device(config-if)# exit	Exits interface configuration mode.

## Configuring GEC over L2TPv3

Perform this task to configure Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface Loopback0**
4. **ip address** *ip-address*
5. **exit**
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation l2tpv3**
8. **ip local interface** *interface-name*
9. **exit**
10. **interface port-channel** *channel-number*
11. **xconnect** *peer-ip-address* **encapsulation l2tpv3** **pw-class** *pw-class-name*
12. **exit**
13. **interface gigabitethernet** *interface-type-number*
14. **channel-group** *channel-group-number*
15. **end**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface Loopback0</b> <b>Example:</b> Device(config)# interface Loopback0	Creates a loopback interface and enters interface configuration mode.
<b>Step 4</b>	<b>ip address ip-address</b> <b>Example:</b> Device(config-if)# ip address 10.1.0.1 255.255.255.255	Assigns an IP address to the interface.
<b>Step 5</b>	<b>exit</b> <b>Example:</b> Device(config-if)# exit	Exits interface configuration mode.
<b>Step 6</b>	<b>pseudowire-class [pw-class-name]</b> <b>Example:</b> Device(config)# pseudowire-class l2tpv3	Enters pseudowire class configuration mode and optionally specifies the name of the Layer 2 Tunneling Protocol (L2TP) pseudowire class.
<b>Step 7</b>	<b>encapsulation l2tpv3</b> <b>Example:</b> Device(config-pw)# encapsulation l2tpv3	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.
<b>Step 8</b>	<b>ip local interface interface-name</b> <b>Example:</b> Device(config-pw)# ip local interface loopback0	Specifies the provider edge (PE) interface whose IP address is to be used as the source IP address for sending tunneled packets. <ul style="list-style-type: none"><li>• Use the same local interface name for all pseudowire classes that are configured between a pair of PE devices.</li></ul> <b>Note</b> This command must be configured for pseudowire-class configurations using L2TPv3 as the data encapsulation method.
<b>Step 9</b>	<b>exit</b> <b>Example:</b> Device(config-pw)# exit	Exits pseudowire class configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 10	<b>interface</b> <b>port-channel</b> <i>channel-number</i>  <b>Example:</b> Device# interface port-channel 1	Defines a port channel and enters interface configuration mode.
Step 11	<b>xconnect</b> <i>peer-ip-address</i> <b>encapsulation l2tpv3</b> <b>pw-class</b> <i>pw-class-name</i>  <b>Example:</b> Device(config-subif)# xconnect 10.0.3.201 1234 encapsulation l2tpv3 pw-class l2tpv3	Specifies the IP address of the peer PE device and the 32-bit virtual circuit identifier (VCI) shared between the PE at each end of the control channel. <ul style="list-style-type: none"> <li>The combination of the peer device ID and the VCI must be unique.</li> <li><b>pw-class</b> <i>pw-class-name</i>—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The <b>pw-class</b> parameter binds <b>xconnect</b> to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.</li> </ul>
Step 12	<b>exit</b>  <b>Example:</b> Device(config-subif)# exit	Exits subinterface configuration mode and enters global configuration mode.
Step 13	<b>interface</b> <b>gigabitEthernet</b> <i>interface-type-number</i>  <b>Example:</b> Device(config)# interface gigabitEthernet 0/0/0	Enters interface configuration mode.
Step 14	<b>channel-group</b> <i>channel-group-number</i>  <b>Example:</b> Device(config-if)# channel-group 1	Add the interface to an EtherChannel group.
Step 15	<b>end</b>  <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

## Configuring GEC with Dot1Q

Perform this task to configure Gigabit EtherChannel (GEC) with VLAN over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *interface-number*
4. **encapsulation dot1q** *vlan-id*
5. **xconnect** *peer-ip-address* **encapsulation l2tpv3** **pw-class** *pw-class-name*

## 6. end

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface port-channel interface-number</b> <b>Example:</b> Device(config)# interface port-channel 1.1	Defines a port channel and enters subinterface configuration mode.
<b>Step 4</b>	<b>encapsulation dot1q vlan-id</b> <b>Example:</b> Device(config-subif)# encapsulation dot1q 100	Specifies that dot1q is used as the data encapsulation method to tunnel IP traffic.
<b>Step 5</b>	<b>xconnect peer-ip-address encapsulation l2tpv3 pw-class pw-class-name</b> <b>Example:</b> Device(config-subif)# xconnect 10.0.3.201 1234 encapsulation l2tpv3 pw-class l2tpv3	Specifies the IP address of the peer provider edge (PE) device and the 32-bit virtual circuit identifier (VCI) that is shared between the PE device at each end of the control channel. <ul style="list-style-type: none"><li>• The combination of the peer device ID and the VCI must be unique.</li><li>• <b>pw-class pw-class-name</b>—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The <b>pw-class</b> parameter binds <b>xconnect</b> to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.</li></ul>
<b>Step 6</b>	<b>end</b> <b>Example:</b> Device# end	Exits subinterface configuration mode and returns to privileged EXEC mode .

## Configuring GEC with QinQ

Perform this task to configure Gigabit EtherChannel (GEC) with queue-in-queue (QinQ) over Layer 2 Tunneling Protocol Version 3 (L2TPv3).

## SUMMARY STEPS

## 1. enable



2. **configure terminal**
3. **interface port-channel** *interface-number*
4. **encapsulation dot1q** *vlan-id* **second-dot1q** *second-vlan-id*
5. **xconnect** *peer-ip-address* **encapsulation l2tpv3** **pw-class** *pw-class-name*
6. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>interface port-channel</b> <i>interface-number</i> <b>Example:</b> Device(config)# interface port-channel 1.1	Defines the subinterface as a port channel and enters subinterface configuration mode.
Step 4	<b>encapsulation dot1q</b> <i>vlan-id</i> <b>second-dot1q</b> <i>second-vlan-id</i> <b>Example:</b> Device(config-subif)# encapsulation dot1q 100 second-dot1q 200	Specifies that QinQ is used as the data encapsulation method to tunnel IP traffic.
Step 5	<b>xconnect</b> <i>peer-ip-address</i> <b>encapsulation l2tpv3</b> <b>pw-class</b> <i>pw-class-name</i> <b>Example:</b> Device(config-subif)# xconnect 10.0.3.202 1234 encapsulation l2tpv3 pw-class l2tpv3	Specifies the IP address of the peer provider edge (PE) device and the 32-bit virtual circuit identifier (VCI) that is shared between the PE device at each end of the control channel. <ul style="list-style-type: none"> <li>• The combination of the peer device ID and the VCI must be unique.</li> <li>• <b>pw-class</b> <i>pw-class-name</i>—The pseudowire class configuration from which the data encapsulation type (L2TPv3) is taken. The <b>pw-class</b> parameter binds <b>xconnect</b> to a specific pseudowire class. The pseudowire class then serves as a template for all attachment circuits bound to it.</li> </ul>
Step 6	<b>end</b> <b>Example:</b> Device# end	Exits subinterface configuration mode and returns to privileged EXEC 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
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>clear l2tun</b> { <b>l2tp-class</b> <i>l2tp-class-name</i>   <b>tunnel id</b> <i>tunnel-id</i>   <b>local ip</b> <i>ip-address</i>   <b>remote ip</b> <i>ip-address</i>   <b>all</b> } <b>Example:</b> Device# clear l2tun tunnel id 56789	Clears the specified L2TPv3 tunnel. (This command is not available if there are no L2TPv3 tunnel sessions configured.) <ul style="list-style-type: none"> <li>• <b>l2tp-class</b> <i>l2tp-class-name</i>—All L2TPv3 tunnels with the specified L2TP class name are torn down.</li> <li>• <b>tunnel id</b> <i>tunnel-id</i>—The L2TPv3 tunnel with the specified tunnel ID are torn down.</li> <li>• <b>local ip</b> <i>ip-address</i>—All L2TPv3 tunnels with the specified local IP address are torn down.</li> <li>• <b>remote ip</b> <i>ip-address</i>—All L2TPv3 tunnels with the specified remote IP address are torn down.</li> <li>• <b>all</b>—All L2TPv3 tunnels are torn down.</li> </ul>

## 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
```

## Example: Verifying an L2TPv3 Session

```

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 device, use the **show l2tun session brief** command.

```

Device# 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 device, use the **show l2tun session all** command.

```

Device# show l2tun session all
L2TP Session Information Total tunnels 1 sessions 1
Session id 2391726297 is up, logical session id 36272, tunnel id 2382731778
  Remote session id is 193836624, remote tunnel id 2280318174
  Locally initiated session
  Unique ID is 12
Session Layer 2 circuit, type is Ethernet, name is GigabitEthernet0/2/0
  Session vcid is 100
  Circuit state is UP
    Local circuit state is UP
    Remote circuit state is UP
  Call serial number is 98300002
  Remote tunnel name is l2tp-asr-2
    Internet address is 6.6.6.6
  Local tunnel name is l2tp-asr-1
    Internet address is 3.3.3.3
IP protocol 115
  Session is L2TP signaled
  Session state is established, time since change 00:05:25
    94 Packets sent, 58 received
    9690 Bytes sent, 5642 received
  Last clearing of counters never
  Counters, ignoring last clear:
    94 Packets sent, 58 received
    9690 Bytes sent, 5642 received
  Receive packets dropped:
    out-of-order:      0
    other:              0
    total:              0
  Send packets dropped:
    exceeded session MTU: 0
    other:              0
    total:              0
  DF bit off, ToS reflect disabled, ToS value 0, TTL value 255
  Sending UDP checksums are disabled
  Received UDP checksums are verified
  No session cookie information available
  FS cached header information:
    encap size = 24 bytes

```

```

45000014 00000000 ff73a965 03030303
06060606 0b8db650
Sequencing is off
Conditional debugging is disabled
SSM switch id is 4101, SSM segment id is 12294

```

## Example: Verifying an L2TP Control Channel

The L2TP control channel is used to negotiate capabilities, monitor the health of the peer PE device, and set up various components of an L2TPv3 session.

To display information about L2TP control channels to other L2TP-enabled devices for all L2TP sessions on the device, use the **show l2tun tunnel** command.

```

Device# 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 about L2TP control channels to other L2TP-enabled devices for all L2TP sessions on the device, use the **show l2tun tunnel all** command.

```

Device# show l2tun tunnel all
L2TP Tunnel Information Total tunnels 1 sessions 1
Tunnel id 2382731778 is up, remote id is 2280318174, 1 active sessions
  Locally initiated tunnel
  Tunnel state is established, time since change 00:02:59
  Tunnel transport is IP (115)
  Remote tunnel name is l2tp-asr-2
    Internet Address 6.6.6.6, port 0
  Local tunnel name is l2tp-asr-1
    Internet Address 3.3.3.3, port 0
  L2TP class for tunnel is l2tp_default_class
  Counters, taking last clear into account:
    54 packets sent, 35 received
    5676 bytes sent, 3442 received
  Last clearing of counters never
  Counters, ignoring last clear:
    54 packets sent, 35 received
    5676 bytes sent, 3442 received
Control Ns 5, Nr 4
  Local RWS 1024 (default), Remote RWS 1024
  Control channel Congestion Control is disabled
  Tunnel PMTU checking disabled
  Retransmission time 1, max 1 seconds
  Unsent queue size 0, max 0
  Resend queue size 0, max 2
  Total resends 0, ZLB ACKs sent 2
  Total out-of-order dropped pkts 0
  Total out-of-order reorder pkts 0
  Total peer authentication failures 0
  Current no session pak queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0
  Control message authentication is disabled

```

## Example: Configuring L2TPv3 Control Channel Authentication

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

**Example: Configuring L2TPv3 Digest Secret Graceful Switchover**

```
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
```

## Example: Configuring Protocol Demultiplexing for L2TPv3

The following example shows how to configure the L2TPv3 Protocol Demultiplexing feature on IPv4 PE devices. The PE devices facing the IPv6 network do not require the IPv6 configuration.

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

## 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
```

## Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport

### Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Dynamic Mode

The following example shows how to configure xconnect on a serial interface with HDLC encapsulation on a dynamic mode. The dynamic mode uses L2TPv3 signaling in control channel to set up the L2TPv3 tunnel.

**Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Static Mode**

```

pseudowire-class 774
 encapsulation l2tpv3
 protocol l2tpv3
 ip local interface GigabitEthernet0/0/1.774
 !
interface Serial0/2/0:0
 no ip address
 xconnect 4.4.4.4 200 pw-class 774

```

**Example: Configuring an L2TPv3 HDLC Like-to-Like Layer 2 Transport on Static Mode**

The following example shows how to configure xconnect on a serial interface with HDLC encapsulation on a static mode. The static mode is used to disable signaling in the L2TPv3 control channel. Since signaling is disabled, you must specify the manual option in xconnect and configure L2TP-specific parameters to complete the L2TPv3 control channel configuration.

```

pseudowire-class pel-ether-pw
 encapsulation l2tpv3
 protocol none
 ip local interface Loopback1
 !
interface Serial0/2/0:0
 no ip address
 xconnect 2.2.2.2 50 encapsulation l2tpv3 manual pw-class pel-ether-pw
 l2tp id 111 111
 l2tp cookie local 4 54321
 l2tp cookie remote 4 12345

```

**Example: Configuring GEC over L2TPv3**

The following is a sample configuration of Gigabit EtherChannel (GEC) over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```

Device> enable
Device# configure terminal
Device(config)# interface Loopback0
Device(config-if)# ip address 10.1.0.1 255.255.255.255
Device(config-if)# exit
Device(config)# pseudowire-class l2tpv3
Device(config-pw)# encapsulation l2tpv3
Device(config-pw)# ip local interface loopback0
Device(config-if)# exit
Device(config)# interface port-channel 1
Device(config-if)# xconnect 1.1.1.1 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-if)# exit
Device(config)# interface g0/0/0
Device(config-if)# channel-group 1
Device(config-if)# end

```

**Example: Configuring GEC with Dot1q over L2TPv3**

The following is a sample configuration of a Gigabit EtherChannel (GEC) with dot1q over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```

Device> enable
Device# configure terminal
Device(config)# interface port-channel 1
Device(config-if)# interface port-channel 1.1

```



```
Device(config-subif)# encapsulation dot1q 100
Device(config-subif)# xconnect 10.0.0.2 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-subif)# end
```

## Example: Configuring GEC with QinQ over L2TPv3

The following is a sample configuration of a Gigabit EtherChannel (GEC) with queue-in-queue (QinQ) over Layer 2 Tunneling Protocol Version 3 (L2TPv3):

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 1
Device(config-if)# interface port-channel 1.1
Device(config-subif)# encapsulation dot1q 100 second-dot1q 200
Device(config-subif)# xconnect 10.0.0.3 1234 encapsulation l2tpv3 pw-class l2tpv3
Device(config-subif)# end
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Master Commands List, All Releases</a>
WAN commands: complete command syntax, command mode, defaults, usage guidelines and examples	<a href="#">Wide-Area Networking Command Reference</a>
Layer 2 Tunnel Protocol Version 3	<i>Layer 2 Tunneling Protocol Version 3</i>
Any Transport over MPLS	<i>Any Transport over MPLS</i>
Cisco 12000 series routers hardware support	<i>Cross-Platform Release Notes for Cisco IOS Release 12.0S</i>
Cisco 7600 series routers hardware support	<i>Cross-Platform Release Notes for Cisco IOS Release 12.2SR</i>
Cisco 3270 series routers hardware support	<i>Release Notes for Cisco IOS Software Release 12.2SE</i>

### Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	<i>Layer Two Tunneling Protocol (Version 3) 'L2TPv3'</i>
draft-martini-l2circuit-trans-mpls-09.txt	<i>Transport of Layer 2 Frames Over MPLS</i>
draft-ietf-pwe3-frame-relay-03.txt	<i>Encapsulation Methods for Transport of Frame Relay over MPLS Networks</i>

Standard/RFC	Title
draft-martini-l2circuit-encap-mpls-04.txt.	<i>Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks</i>
draft-ietf-pwe3-ethernet-encap-08.txt.	<i>Encapsulation Methods for Transport of Ethernet over MPLS Networks</i>
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	<i>Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks</i>
draft-ietf-ppvpn-l2vpn-00.txt.	<i>An Architecture for L2VPNs</i>

### 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:  <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

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

Feature Name	Releases	Feature Information
Layer 2 Tunneling Protocol Version 3	XE 2.6 XE 2.6.2 XE 3.3S XE 3.11S	<p>The Layer 2 Tunneling Protocol Version 3 (L2TPv3) feature expands Cisco's support of Layer 2 VPNs.</p> <p>In Cisco IOS XE Release 2.6, the following features were added:</p> <ul style="list-style-type: none"> <li>• Ethernet over L2TPv3</li> <li>• Layer 2 VPN (L2VPN): Syslog, SNMP Trap, and show Command Enhancements for ATOM and L2TPv3</li> <li>• L2TPv3 Control Message Hashing</li> <li>• L2TPv3 Control Message Rate Limiting</li> <li>• L2TPv3 Digest Secret Graceful Switchover</li> <li>• L2TPv3 Protocol Demultiplexing</li> <li>• L2TPv3 Custom Ethertype for Dot1q and QinQ Encapsulations</li> </ul> <p>In Cisco IOS XE Release 2.6.2, support was added for the <b>ip pmtu</b> command.</p> <p>In Cisco IOS XE Release 3.3S, support for HDLC over L2TPv3 was added.</p> <p>The following commands were introduced or modified: <b>clear l2tun</b>, <b>debug vpdn</b>, <b>ip pmtu</b>, <b>l2tp cookie local</b>, <b>l2tp cookie remote l2tp hello</b>, <b>l2tp id</b>, and <b>xconnect</b>.</p>

## 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 4

# 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, on page 61](#)
- [Prerequisites for L2VPN Pseudowire Redundancy, on page 61](#)
- [Restrictions for L2VPN Pseudowire Redundancy, on page 62](#)
- [Information About L2VPN Pseudowire Redundancy, on page 62](#)
- [How to Configure L2VPN Pseudowire Redundancy, on page 64](#)
- [Configuration Examples for L2VPN Pseudowire Redundancy, on page 74](#)
- [Configuration Examples for L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature, on page 77](#)
- [Additional References, on page 81](#)
- [Feature Information for L2VPN Pseudowire Redundancy, on page 82](#)

## 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](http://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).
  - Any Transport over MPLS
  - L2 VPN Interworking
  - Layer 2 Tunneling Protocol Version 3 (L2TPv3)

- 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

- 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.
- L2VPN Pseudowire Redundancy does not support pseudowire interworking mode with L2TPv3. The connectivity between CEs may be impacted if you have interworking IP configured in the pseudowire class.
- The primary and backup pseudowires must run the same type of transport service. The primary and backup pseudowires must be configured with AToM or L2TPv3.
- The backup peer can only be configured for nonstatic L2TPv3 sessions. The backup L2TPv3 session cannot be static L2TPv3 session. The encapsulation type of primary and backup pseudowire must be the same.
- If you use L2VPN Pseudowire Redundancy with L2VPN Interworking, the interworking method must be the same for the primary and backup pseudowires.
- L2VPN Pseudowire Redundancy does support setting the experimental (EXP) bit on the Multiprotocol Label Switching (MPLS) pseudowire.
- L2VPN Pseudowire Redundancy does not support different pseudowire encapsulation types on the MPLS pseudowire.
- 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.

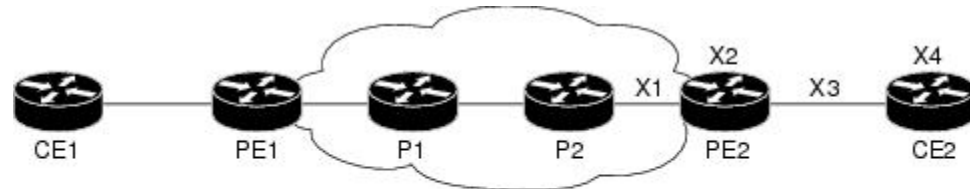
## Information About L2VPN Pseudowire Redundancy

### Introduction to L2VPN Pseudowire Redundancy

L2VPNs can provide pseudowire resiliency through their routing protocols. When connectivity between end-to-end PE 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 figure below shows those parts of the network that are vulnerable to an interruption in service.

**Figure 5: Points of Potential Failure in an L2VPN Network**



X1 = End-to-end routing failure  
 X2 = PE hardware or software failure  
 X3 = Attachment circuit failure from a line break  
 X4 = CE hardware or software failure

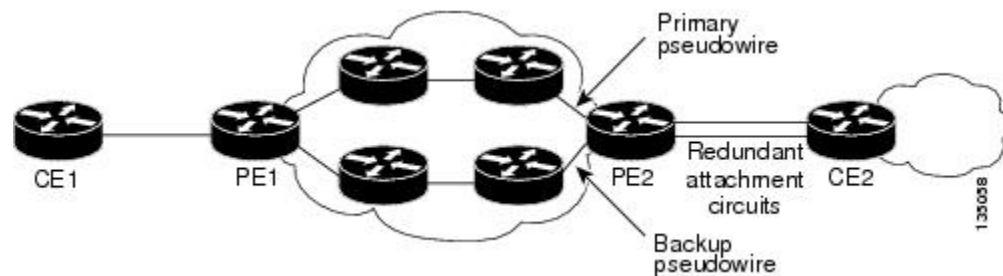
133037

The L2VPN Pseudowire Redundancy feature provides the ability to ensure that the CE2 router 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 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.

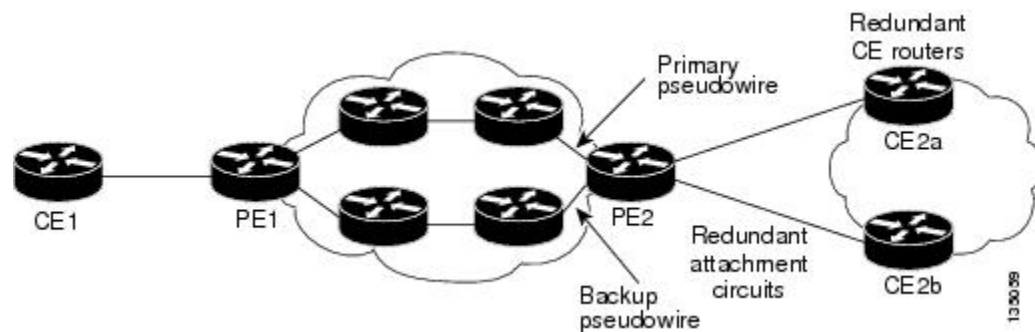
**Figure 6: L2VPN Network with Redundant PWs and Attachment Circuits**



133038

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

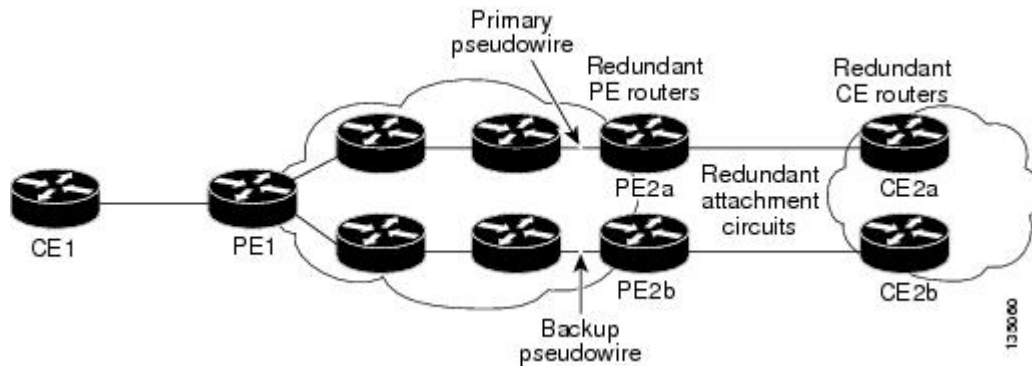
**Figure 7: L2VPN Network with Redundant PWs, Attachment Circuits, and CE Routers**



133039

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

Figure 8: L2VPN Network with Redundant PWs, Attachment Circuits, CE Routers, and PE Routers



## 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.

### 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	<b>enable</b> <b>Example:</b> <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<b>pseudowire-class name</b> <b>Example:</b> <pre>Router(config)# pseudowire-class atom</pre>	Establishes a pseudowire class with a name that you specify. Enters pseudowire class configuration mode.
Step 4	<b>encapsulation mpls</b> <b>Example:</b> <pre>Router(config-pw-class)# encapsulation mpls</pre>	Specifies the tunneling encapsulation. For AToM, the encapsulation type is <b>mpls</b> .
Step 5	<b>interworking {ethernet   ip}</b> <b>Example:</b> <pre>Router(config-pw-class)# interworking ip</pre>	(Optional) Enables the translation between the different Layer 2 encapsulations.

## Configuring the Pseudowire using the commands associated with the L2VPN Protocol-Based CLIs feature

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 **l2vpn xconnect context** command, you receive the following error:

```
% Incomplete command.
```

Perform this task to configure a pseudowire class.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encapsulation mpls**
5. **neighbor** *peer-address* *vcid-value*
6. **interworking** {*ethernet* | *ip*}

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>interface pseudowire</b> <i>number</i> <b>Example:</b> <pre>Router(config)# interface pseudowire 1</pre>	Establishes an interface pseudowire with a value that you specify. Enters pseudowire configuration mode.
<b>Step 4</b>	<b>encapsulation mpls</b> <b>Example:</b> <pre>Router(config-pw)# encapsulation mpls</pre>	Specifies the tunneling encapsulation. For AToM, the encapsulation type is <b>mpls</b> .
<b>Step 5</b>	<b>neighbor</b> <i>peer-address</i> <i>vcid-value</i> <b>Example:</b> <pre>Router(config-pw)# neighbor 10.0.0.1 123</pre>	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
<b>Step 6</b>	<b>interworking</b> { <i>ethernet</i>   <i>ip</i> } <b>Example:</b> <pre>Router(config-pw)# interworking ip</pre>	(Optional) Enables the translation between the different Layer 2 encapsulations.

**Configuring L2VPN Pseudowire Redundancy**

Perform this task 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** *e nable-delay {disable-delay | never}*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>interface gigabitethernet</b> <i>slot / subslot / interface . subinterface</i> <b>Example:</b> <pre>Router(config)# interface gigabitethernet0/0/0.1</pre>	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. <b>Note</b> Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
<b>Step 4</b>	<b>encapsulation dot1q</b> <i>vlan-id</i> <b>Example:</b> <pre>Router(config-subif)# encapsulation dot1q 100</pre>	Enables the subinterface to accept 802.1Q VLAN packets. <b>Note</b> The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet.
<b>Step 5</b>	<b>xconnect</b> <i>peer-router-id vcid {encapsulation mpls  pw-class pw-class-name}</i> <b>Example:</b> <pre>Router(config-subif)# xconnect 10.0.0.1 123 pw-class atom</pre>	Binds the attachment circuit to a pseudowire VC and enters xconnect configuration mode. <ul style="list-style-type: none"> <li>• The syntax for this command is the same as for all other Layer 2 transports.</li> </ul>

	Command or Action	Purpose
<b>Step 6</b>	<p><b>backup peer</b> <i>peer-router-ip-addr vcid</i> [<b>pw-class</b> <i>pw-class-name</i>]</p> <p><b>Example:</b></p> <pre>Router(config-if-xconn)# backup peer 10.0.0.3 125 pw-class atom</pre>	<p>Specifies a redundant peer for the pseudowire VC.</p> <p>The pseudowire class name must match the name that you specified when you created the pseudowire class, but you can use a different pw-class in the <b>backup peer</b> command than the name that you used in the primary <b>xconnect</b> command.</p>
<b>Step 7</b>	<p><b>backup delay</b> <i>e nable-delay {disable-delay   never}</i></p> <p><b>Example:</b></p> <pre>Router(config-if-xconn)# backup delay 5 never</pre>	<p>Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is from 0 to 180.</p> <p>Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is from 0 to 180 seconds. If you specify the <b>never</b> keyword, the primary pseudowire VC never takes over for the backup.</p>

## Configuring L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature

Perform this task to configure the L2VPN Pseudowire Redundancy feature.

### Before you begin

For each transport type, the **l2vpn xconnect context** 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 **l2vpn xconnect context** 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. **end**
6. **interface pseudowire** *number*
7. **source template type pseudowire** *template-name*
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **member pseudowire** *interface-number*
12. **member pseudowire** *interface-number*
13. **member gigabitethernet** *interface-number*
14. **redundancy delay** *enable-delay {disable-delay | never}*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface gigabitethernet slot / subslot / interface . subinterface</b> <b>Example:</b> Device(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.
<b>Step 4</b>	<b>encapsulation dot1q vlan-id</b> <b>Example:</b> Device(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.
<b>Step 5</b>	<b>end</b> <b>Example:</b> Router(config-subif)# end	Exits to privileged EXEC mode.
<b>Step 6</b>	<b>interface pseudowire number</b> <b>Example:</b> Router(config)# interface pseudowire 100	Specifies the pseudowire interface and enters interface configuration mode.
<b>Step 7</b>	<b>source template type pseudowire template-name</b> <b>Example:</b> Router(config-if)# source template type pseudowire atom	Configures the source template of type pseudowire named atom
<b>Step 8</b>	<b>neighbor peer-address vcid-value</b> <b>Example:</b> Router(config-if)# neighbor 10.0.0.1 123	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
<b>Step 9</b>	<b>exit</b> <b>Example:</b> Router(config-if)# exit	Exits to privileged EXEC mode.

	Command or Action	Purpose
<b>Step 10</b>	<b>l2vpn xconnect context</b> <i>context-name</i> <b>Example:</b> <pre>Router(config)# l2vpn xconnect context con1</pre>	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
<b>Step 11</b>	<b>member pseudowire</b> <i>interface-number</i> <b>Example:</b> <pre>Device(config-xconnect)# member pseudowire 100 group GR_1 priority 2</pre>	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
<b>Step 12</b>	<b>member pseudowire</b> <i>interface-number</i> <b>Example:</b> <pre>Device(config-xconnect)# member pseudowire 1001 group GR_1 priority 2</pre>	Specifies a second member pseudowire for redundancy.
<b>Step 13</b>	<b>member gigabitethernet</b> <i>interface-number</i> <b>Example:</b> <pre>Device(config-xconnect)# member GigabitEthernet0/0/0.1 service instance 1</pre>	Specifies the location of the Gigabit Ethernet member interface.
<b>Step 14</b>	<b>redundancy delay</b> <i>enable-delay</i> { <i>disable-delay</i>   <b>never</b> } <b>Example:</b> <pre>Device(config-xconnect)# redundancy delay 0 0 group GR_1</pre>	<p>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.</p> <p>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 <b>never keyword</b>, the primary pseudowire VC never takes over for the backup.</p>

## 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 moves to the fully active state when executing the command.

### SUMMARY STEPS

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

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>xconnect backup force-switchover { interface interface-info   peer ip-address vcid}</b> <b>Example:</b> <pre>Router# xconnect backup force-switchover peer 10.10.10.1 123</pre>	Specifies that the router should switch to the backup or to the primary pseudowire.

## Verifying the L2VPN Pseudowire Redundancy Configuration

Perform this task 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

The following is sample output from the **show mpls l2transport vc** command. In this example, the primary attachment circuit is up. The backup attachment circuit is available, but not currently selected.

**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





2. `show l2vpn service all`
3. `logging redundancy`
4. `logging pseudowire status`

## DETAILED STEPS

### Step 1 `show l2vpn atom 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:

```
Device# show l2vpn atom 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 l2vpn atom 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 l2vpn service all`

In this example, the topology is attachment circuit 1 to pseudowire 1 with apPseudowire 2 as a backup:

#### Example:

```
Device# show l2vpn service all
Legend: St=State      XC St=State in the L2VPN Service      Prio=Priority
        UP=Up         DN=Down                               AD=Admin Down      IA=Inactive
        SB=Standby   HS=Hot Standby                       RV=Recovering      NH=No Hardware
        m=manually selected

Interface          Group          Encapsulation          Prio  St  XC St
-----
VPWS name: foo, State: UP
Eth1/1.1
pw101              blue          102.1.1.1:100 (MPLS)   0     UP  UP
pw102              blue          103.1.1.1:100 (MPLS)   5     SB  IA
pw103              blue          104.1.1.1:100 (MPLS)   8     SB  IA
pw104              blue          105.1.1.1:100 (MPLS)  11    SB  IA
```

In this example, the topology is attachment circuit 1 to attachment circuit 2 with a pseudowire backup for attachment circuit 2:

#### Example:

```
Device# show l2vpn service 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 logging redundancy

In addition to the **show l2vpn atom vc** command and the **show l2vpn service** command, you can use the **logging redundancy** command to enable system message log (syslog) reporting of xconnect redundancy status events:

#### Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging redundancy
```

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

Activating the primary member:

#### Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging pseudowire status
```

When this command is configured, this is configured the status of the pseudowire can be monitored:

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
```

### Step 4 logging pseudowire status

you can use the **logging pseudowire status** command to monitor the status of the pseudowire.

#### Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging pseudowire status
```

## 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
```

## Example L2VPN Pseudowire Redundancy and AToM (Like to Like)

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
```

## Example L2VPN Pseudowire Redundancy and L2VPN Interworking

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
```

## Example L2VPN Pseudowire Redundancy with Layer 2 Local Switching

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
```

## Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3

The following example shows how to configure a backup peer for an xconnect session:

```
pseudowire-class 773
  encapsulation l2tpv3
  ip local interface GigabitEthernet0/0/0.773
  !
pseudowire-class 774
  encapsulation l2tpv3
  ip local interface GigabitEthernet0/0/1.774
  !
interface GigabitEthernet0/0/0.780
  encapsulation dot1Q 780
  xconnect 10.22.73.14 100 pw-class 773
  backup peer 10.22.74.14 101 pw-class 774
  backup delay 0 0
```

The following example shows how to configure a Gigabit Ethernet port with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/2
  xconnect 10.22.70.83 50 pw-class pe1-pw-primary
  backup peer 20.22.70.85 51 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet VLAN with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.100
  encapsulation dot1q 100
  xconnect 10.22.70.83 60 pw-class pe1-pw-primary
  backup peer 10.22.70.85 61 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet Q-in-Q with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.200
  encapsulation dot1q 200 second-dot1q 400
```

```
xconnect 10.22.70.83 70 pw-class pe1-pw-primary
backup peer 10.22.70.85 71 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet Q-in-any with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.300
 encapsulation dot1q 300 second-dot1q any
 xconnect 10.22.70.83 80 pw-class pe1-pw-primary
 backup peer 10.22.70.85 81 pw-class pe1-pw-secondary
```

The following example shows how to configure an HDLC with L2VPN pseudowire redundancy and L2TPv3

```
interface Serial10/2/0:0
 no ip address
 xconnect 10.22.71.83 40 pw-class pe1-pw-hdlc
 backup peer 10.22.70.85 41 pw-class pe1-pw-hdlc-2
```

## Configuration Examples for L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature

Each of the configuration examples refers to one of the following interface pseudowires:

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

```
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
```

- L2VPN IP interworking:

```
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
 interworking ip
```

### Example L2VPN Pseudowire Redundancy and AToM (Like to Like) using the commands associated with the L2VPN Protocol-Based CLIs feature

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

```
interface Serial14/0
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.55.55.3 4001
 !
 l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
```

```
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
```

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

```
connect fr-fr-pw Serial6/0 225 l2transport
interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.55.55.3 5226
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
```

## Example L2VPN Pseudowire Redundancy and L2VPN Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature

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

```
interface Ethernet0/0
interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking 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
 interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking 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
interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
```

```

member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

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

```

interface Serial17/0
 encapsulation ppp
 interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip

```

## Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3 using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure a backup peer for an xconnect session:

```

interface pseudowire 773
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/0.773
!
interface pseudowire 774
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/1.774
!
interface GigabitEthernet0/0/0.780
 encapsulation dot1Q 780
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.73.14 100
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip

```

The following example shows how to configure a Gigabit Ethernet port with L2VPN pseudowire redundancy and L2TPv3:

```

interface GigabitEthernet0/0/2
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.70.83 50
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2

```

```

member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

The following example shows how to configure a Gigabit Ethernet VLAN with L2VPN pseudowire redundancy and L2TPv3:

```

interface GigabitEthernet0/0/0.100
encapsulation dot1q 100
interface pseudowire 100
source template type pseudowire ether-pw
neighbor 10.22.70.83 60
!
l2vpn xconnect context con1
member pseudowire 100 group GR_1 priority 1
member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

The following example shows how to configure a Gigabit Ethernet Q-in-Q with L2VPN pseudowire redundancy and L2TPv3:

```

interface GigabitEthernet0/0/0.200
encapsulation dot1q 200 second-dot1q 400
interface pseudowire 100
source template type pseudowire ether-pw
neighbor 10.22.70.83 70
!
l2vpn xconnect context con1
member pseudowire 100 group GR_1 priority 1
member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

The following example shows how to configure a Gigabit Ethernet Q-in-any with L2VPN pseudowire redundancy and L2TPv3:

```

interface GigabitEthernet0/0/0.300
encapsulation dot1q 300 second-dot1q any
interface pseudowire 100
source template type pseudowire ether-pw
neighbor 10.22.70.83 80
!
l2vpn xconnect context con1
member pseudowire 100 group GR_1 priority 1
member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

The following example shows how to configure an HDLC with L2VPN pseudowire redundancy and L2TPv3

```

interface Serial0/2/0:0
no ip address
interface pseudowire 100
source template type pseudowire ether-pw
neighbor 10.22.71.83 40
!
l2vpn xconnect context con1

```



```

l2vpn xconnect context con1
member pseudowire 100 group GR_1 priority 1
member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip

```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
Wide-area networking commands	<i>Cisco IOS Wide-Area Networking Command Reference</i>
Cisco IOS XE Multiprotocol Label Switching configuration tasks	<i>Cisco IOS XE Multiprotocol Label Switching Configuration Guide</i>
Cisco IOS XE Wide-area networking configuration tasks	<i>Cisco IOS XE Wide-Area Networking Configuration Guide</i>

### Standards

Standards	Title
None	--

### 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 software releases, and feature sets, use Cisco MIB Locator found at the following URL:  <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

### RFCs

RFCs	Title
None	--

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 4: Feature Information for L2VPN Pseudowire Redundancy**

Feature Name	Releases	Feature Information
L2VPN Pseudowire Redundancy	XE 2.3 XE 3.3S	<p>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.</p> <p>In Cisco IOS XE Release 2.3, this feature was integrated into the Cisco ASR 1000 Series Aggregation Service Routers.</p> <p>In Cisco IOS XE Release 3.3S, this feature supports Layer 2 Tunneling Protocol Version 3 (L2TPv3).</p> <p>The following commands were introduced or modified: <b>backup delay (L2VPN local switching), backup peer, show xconnect, xconnect backup force-switchover, xconnect logging redundancy.</b></p>
L2VPN Pseudowire Redundancies	Cisco IOS XE Fuji 16.9.1	In Cisco IOS XE Fuji 16.9.1, this feature is supported on Cisco 1000 Series ISRs.



## 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 following interface-to-interface switching combinations are supported by this feature:

- ATM to ATM
- ATM to Ethernet
- Ethernet/Ethernet VLAN to Ethernet/Ethernet VLAN
- Frame Relay to Frame Relay

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, on page 83](#)
- [Prerequisites for Layer 2 Local Switching, on page 84](#)
- [Restrictions for Layer 2 Local Switching, on page 84](#)
- [Information About Layer 2 Local Switching, on page 84](#)
- [How to Configure Layer 2 Local Switching, on page 85](#)
- [Configuration Examples for Layer 2 Local Switching, on page 94](#)
- [Additional References, on page 98](#)
- [Feature Information for Layer 2 Local Switching, on page 99](#)

## 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](http://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 ASR 1000 Series Aggregation Services Router.

## Restrictions for Layer 2 Local Switching

- For Ethernet/Ethernet VLAN circuits, the Cisco ASR 1000 Series Aggregation Services Router must have Ethernet Adapters.
- For Frame Relay local switching, you must globally issue the **frame-relay switching** command.

## 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, Ethernet to Ethernet or Frame Relay to Frame Relay) or between interfaces of different types (for example, Ethernet VLAN to Ethernet VLAN or Ethernet to Ethernet VLAN) 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 the 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 and checkpointing of data to ensure minimal packet loss when the primary Route Processor goes down. NSF/SSO support is available for the following locally switched attachment circuits:

- Ethernet/Ethernet VLAN to Ethernet/Ethernet VLAN
- Frame Relay to Frame Relay

### Layer 2 Local Switching Applications

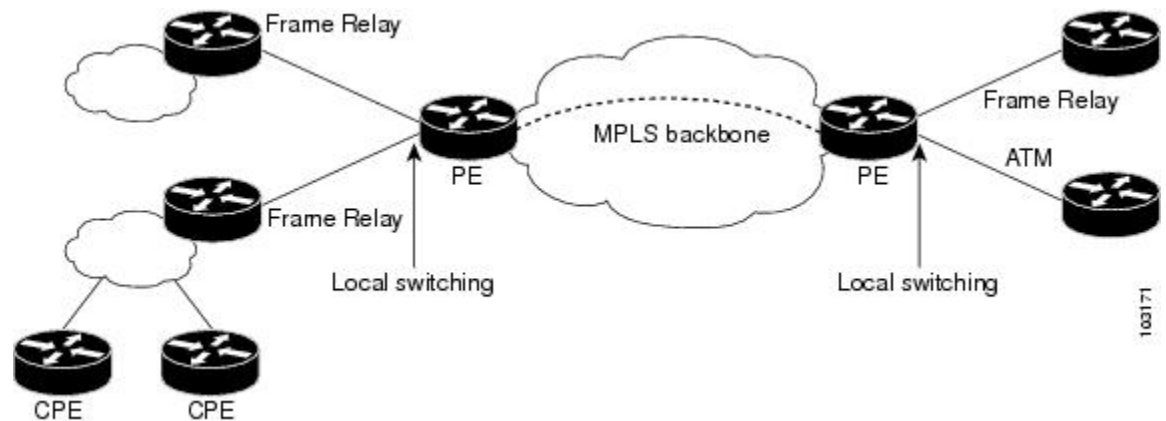
Incumbent local exchange carriers (ILECs) that use an interexchange carrier (IXC) to carry traffic between two local exchange carriers can use the Layer 2 Local Switching feature. Telecom regulations require the ILECs to pay the IXCs to carry that traffic. At times, the ILECs cannot terminate customer 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**



## How to Configure Layer 2 Local Switching

### Configuring Ethernet VLAN Same-Port Switching

Perform this task to configure 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 type number type number**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b> <pre>Router&gt; enable</pre>	<ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>interface fastethernet slot / port . subinterface-number</b> <b>Example:</b> <pre>Router(config)# interface fastethernet6/0.1</pre>	Specifies the first Fast Ethernet line card, subslot (if available), port, and subinterface, and enters subinterface configuration mode.
<b>Step 4</b>	<b>encapsulation dot1q vlan-id</b> <b>Example:</b> <pre>Router(config-subif)# encapsulation dot1q 10</pre>	Enables the subinterface to accept 802.1Q VLAN packets and specifies the first VLAN.
<b>Step 5</b>	<b>exit</b> <b>Example:</b> <pre>Router(config-subif)# exit</pre>	Exits subinterface configuration mode and returns to global configuration mode.
<b>Step 6</b>	<b>interface fastethernet slot / port . subinterface-number</b> <b>Example:</b> <pre>Router(config)# interface fastethernet6/0.2</pre>	Specifies the second Fast Ethernet line card, subslot (if available), port, and subinterface, and enters subinterface configuration mode.
<b>Step 7</b>	<b>encapsulation dot1q vlan-id</b> <b>Example:</b> <pre>Router(config-subif)# encapsulation dot1q 20</pre>	Enables the subinterface to accept 802.1Q VLAN packets and specifies the second VLAN.
<b>Step 8</b>	<b>exit</b> <b>Example:</b> <pre>Router(config-subif)# exit</pre>	Exits subinterface configuration mode and returns to global configuration mode.
<b>Step 9</b>	<b>connect connection-name type number type number</b> <b>Example:</b> <pre>Router(config)# connect conn fastethernet 6/0.1 fastethernet 6/0.2</pre>	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

Perform this task to configure local switching for Ethernet (port mode) to Ethernet VLAN.

### 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 type number type number*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> <pre>Router&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	<b>interface fastethernet</b> <i>slot / subslot / port</i> <b>Example:</b> <pre>Router(config)# interface fastethernet3/0/0</pre>	Specifies a Fast Ethernet line card, subslot (if available), and port, and enters interface configuration mode. <ul style="list-style-type: none"> <li>• This is the interface on one side of the PE router that passes Ethernet packets to and from the customer edge (CE) router.</li> </ul>
Step 4	<b>interface fastethernet</b> <i>slot / port / subinterface-number</i> <b>Example:</b> <pre>Router(config-if)# interface fastethernet6/0/0.1</pre>	Specifies a Fast Ethernet line card, subslot (if available), port, and subinterface, and enters subinterface configuration mode. <ul style="list-style-type: none"> <li>• This is the interface on the other side of the PE router than passes Ethernet VLAN packets to and from the CE router.</li> </ul>
Step 5	<b>encapsulation dot1q</b> <i>vlan-id</i> <b>Example:</b> <pre>Router(config-subif)# encapsulation dot1q 100</pre>	Enables the interface to accept 802.1Q VLAN packets.
Step 6	<b>exit</b> <b>Example:</b>	Exits subinterface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	<code>Router(config-subif)# exit</code>	
<b>Step 7</b>	<p><b>connect</b> <i>connection-name type number type number</i></p> <p><b>Example:</b></p> <pre>Router(config)# connect eth-ethvlan-con fastethernet 3/0/0 fastethernet 6/0/0.1</pre>	Creates a local connection between the two interfaces.

## 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.

### 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
<b>Step 1</b>	<p><b>enable</b></p> <p><b>Example:</b></p> <pre>Router&gt; enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<p><b>configure terminal</b></p> <p><b>Example:</b></p> <pre>Router# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<p><b>interface atm</b> <i>slot / port</i></p> <p><b>Example:</b></p> <pre>Router(config)# interface atm1/0/0</pre>	Specifies an ATM line card, subslot (if available), and port, and enters interface configuration mode.
<b>Step 4</b>	<p><b>pvc</b> <i>vpi / vci l2transport</i></p> <p><b>Example:</b></p> <pre>Router(config-if)# pvc 1/100 l2transport</pre>	<p>Assigns a VPI and VCI and enters ATM PVC l2transport configuration mode.</p> <ul style="list-style-type: none"> <li>• The <b>l2transport</b> keyword indicates that the PVC is a switched PVC instead of a terminated PVC.</li> </ul>



	Command or Action	Purpose
Step 5	<b>encapsulation</b> <i>layer-type</i> <b>Example:</b> Router(cfg-if-atm-l2trans-pvc)# encapsulation aal5	Specifies the encapsulation type for the ATM PVC. Both AAL0 and AAL5 are supported. <ul style="list-style-type: none"> <li>Repeat Steps 3 through 5 for another ATM PVC on the same router.</li> </ul>
Step 6	<b>exit</b> <b>Example:</b> Router(cfg-if-atm-l2trans-pvc)# exit	Exits PVC l2transport configuration mode and returns to interface configuration mode.
Step 7	<b>exit</b> <b>Example:</b> Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 8	<b>connect</b> <i>connection-name interface pvc interface pvc</i> <b>Example:</b> 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 90](#).

### 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	<b>enable</b> <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b>	Enters global configuration mode.

	Command or Action	Purpose
	<code>Router# configure terminal</code>	
<b>Step 3</b>	<b>interface atm slot/port</b> <b>Example:</b> <code>Router(config)# interface atm1/0</code>	Specifies an ATM line card, subslot (if available), and port and enters interface configuration mode.
<b>Step 4</b>	<b>atm pvp vpi l2transport</b> <b>Example:</b> <code>Router(config-if)# atm pvp 100 l2transport</code>	Identifies the virtual path and enters PVP l2transport configuration mode. The <b>l2transport</b> keyword indicates that the PVP is a switched PVP instead of a terminated PVP. <ul style="list-style-type: none"> <li>Repeat Steps 3 and 4 for another ATM permanent virtual path on the same router.</li> </ul>
<b>Step 5</b>	<b>exit</b> <b>Example:</b> <code>Router(config-if-atm-l2trans-pvp)# exit</code>	Exits PVP l2transport configuration mode and returns to interface configuration mode.
<b>Step 6</b>	<b>exit</b> <b>Example:</b> <code>Router(config-if)# exit</code>	Exits interface configuration mode and returns to global configuration mode.
<b>Step 7</b>	<b>connect connection-name interface pvp interface pvp</b> <b>Example:</b> <code>Router(config)# connect atm-con atm1/0 100 atm2/0 200</code>	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
<b>Step 1</b>	<b>enable</b> <b>Example:</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>

	Command or Action	Purpose
	<code>Router&gt; enable</code>	
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <code>Router# configure terminal</code>	Enters global configuration mode.
<b>Step 3</b>	<b>interface atm slot / subslot / port</b> <b>Example:</b> <code>Router(config)# interface atm1/0/0</code>	Specifies an ATM line card, subslot (if available), and port, and enters interface configuration mode.
<b>Step 4</b>	<b>atm pvp vpi l2transport</b> <b>Example:</b> <code>Router(config-if)# atm pvp 100 l2transport</code>	Specifies one VPI and enters PVP l2transport configuration mode. Repeat this step for the other ATM permanent virtual path on this same port. <ul style="list-style-type: none"> <li>• The <b>l2transport</b> keyword indicates that the indicated PVP is a switched PVP instead of a terminated PVP.</li> </ul>
<b>Step 5</b>	<b>exit</b> <b>Example:</b> <code>Router(config-if-atm-l2trans-pvp)# exit</code>	Exits PVP l2transport configuration mode and returns to interface configuration mode.
<b>Step 6</b>	<b>exit</b> <b>Example:</b> <code>Router(config-if)# exit</code>	Exits interface configuration mode and returns to global configuration mode.
<b>Step 7</b>	<b>connect connection-name interface pvp interface pvp</b> <b>Example:</b> <code>Router(config)# connect atm-con atm1/0/0 100 atm1/0/0 200</code>	In global configuration mode, creates the local connection between the two specified permanent virtual paths.

## Configuring Frame Relay-to-Frame Relay Local Switching

For information about Frame Relay-to-Frame Relay local switching, see the Distributed Frame Relay Switching feature module.

### 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
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip cef distributed</b> <b>Example:</b> Router(config)# ip cef distributed	Enables Cisco Express Forwarding operation.
<b>Step 4</b>	<b>frame-relay switching</b> <b>Example:</b> Router(config)# frame-relay switching	Enables PVC switching on a Frame Relay DCE device or a Network-to-Network Interface (NNI).
<b>Step 5</b>	<b>interface</b> <i>type number</i> <b>Example:</b> Router(config)# interface serial 0	Specifies a Frame Relay interface and enters interface configuration mode.
<b>Step 6</b>	<b>encapsulation frame-relay</b> <i>[cisco   ietf]</i> <b>Example:</b> Router(config-if)# encapsulation frame-relay	Enables Frame Relay encapsulation. <ul style="list-style-type: none"><li>• The default is <b>cisco</b> encapsulation.</li><li>• You do not need to specify an encapsulation type.</li></ul>
<b>Step 7</b>	<b>frame-relay interface-dlci</b> <i>dlci switched</i> <b>Example:</b> Router(config-if)# frame-relay interface-dlci 100 switched	(Optional) Creates a switched PVC and enters Frame Relay DLCI configuration mode. <ul style="list-style-type: none"><li>• Repeat Steps 5 through 7 for each switched PVC.</li><li>• If you do not create a Frame Relay PVC in this step, it will automatically be created by the <b>connect</b> command.</li></ul>
<b>Step 8</b>	<b>exit</b> <b>Example:</b> Router(config-fr-dlci)# exit	Exits Frame Relay DLCI configuration mode and returns to interface configuration mode.

	Command or Action	Purpose
Step 9	<b>exit</b> <b>Example:</b> <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 10	<b>connect</b> <i>connection-name interface dlc1 interface dlc1</i> <b>Example:</b> <pre>Router(config)# connect connection1 serial0 100 serial1 101</pre>	Defines a connection between Frame Relay PVCs.

## Verifying Layer 2 Local Switching

### Verifying Layer 2 Local Switching Configuration

To verify configuration of the Layer 2 local switching feature, use the **show connection** command on the provider edge (PE) router.

#### SUMMARY STEPS

1. **show connection** [**all** | *element* | **id** *id* | **name** *name* | **port** *port*]

#### DETAILED STEPS

---

```
show connection [all | element | id id | name name | port port]
```

The **show connection** command displays the local connection between a Gigabit Ethernet interface and another local Gigabit Ethernet interface:

#### Example:

```
Router# show connection name ethconn1
Connection: 1 - ethconn1
Current State: UP
Segment 1: GigabitEthernet0/0/0.1 up
Segment 2: GigabitEthernet0/0/0.2 up
```

---

### 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/Ethernet VLAN to Ethernet/Ethernet VLAN

For information about configuring NSF/SSO on the Route Processors, see the "Stateful Switchover" module in the *Cisco IOS XE High Availability Configuration Guide*. Perform this task to verify that the NSF/SSO: Layer 2 Local Switching feature is working correctly.

## SUMMARY STEPS

1. ping
2. redundancy force-switchover
3. show connection all
4. ping

## DETAILED STEPS

### Step 1 ping

Issue the **ping** command or initiate traffic between the two CE routers.

### Step 2 redundancy force-switchover

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 show connection all

Issue the **show connection all** command to ensure that the Layer 2 local switching connection on the dual RP is operating:

#### Example:

```
Router# show connection all
D   Name           Segment 1           Segment 2           State
-----
1   conn           Gi0/0/0.1          Gi0/0/0.2           UP
```

### Step 4 ping

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 conn**
- **show connection**

# Configuration Examples for Layer 2 Local Switching

## Example: Configuring Ethernet VLAN Same-Port Switching

The following example shows same-port switching between two VLANs on one Ethernet interface:

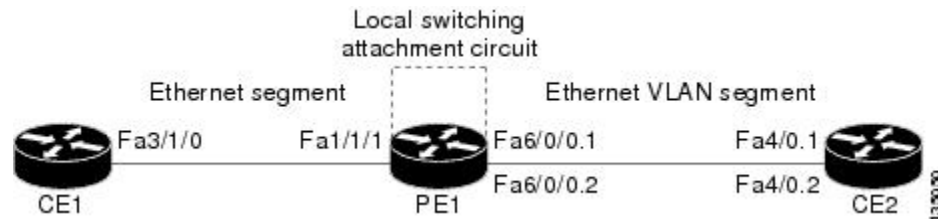
```
interface fastethernet 0/0.1
 encapsulation dot1q 1
 interface fastethernet 0/0.2
```

```
encapsulation dot1q 2
connect conn FastEthernet 0/0.1 FastEthernet 0/0.2
```

## Example: Configuring NSF SSO Ethernet Port Mode to Ethernet VLAN Local Switching

The following configuration uses the network topology shown in the figure below.

**Figure 10: 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
<pre>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</pre>	<pre>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</pre>

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

!

!

ip routing

ip cef distributed

!

interface fa1/1/1

  description - connection to CE1 fa3/1/0

  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
```

## 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: Configuring 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
```

# Additional References

## Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
WAN Commands	<i>Cisco IOS Wide-Area Networking Command Reference</i>
Stateful switchover configuration information	"Stateful Switchover " module in the <i>Cisco IOS XE High Availability Configuration Guide</i>

## Standards

Standard	Title
draft-ietf-l2tpext-l2tp-base-03.txt	<i>Layer Two Tunneling Protocol (Version 3) 'L2TPv3'</i>
draft-martini-l2circuit-trans-mpls-09.txt	<i>Transport of Layer 2 Frames Over MPLS</i>
draft-martini-l2circuit-encap-mpls-04.txt	<i>Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks</i>
draft-ietf-ppvpn-l2vpn-00.txt	<i>An Architecture for L2VPNs</i>

## MIBs

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

## RFCs

RFC	Title
None	--

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 5: Feature Information for Layer 2 Local Switching**

Feature Name	Releases	Feature Information
Layer 2 Local Switching	Cisco IOS XE Release 2.5	<p>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.</p> <p>In Cisco IOS XE Release 2.5, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. Support was added for the following local switching types:</p> <ul style="list-style-type: none"> <li>• Ethernet to Ethernet VLAN</li> <li>• Same-port switching for Ethernet VLAN</li> </ul> <p>The following commands were introduced or modified: <b>connect</b> (L2VPN local switching), <b>show connection</b>.</p>
Layer 2 Local Switching - ATM to ATM	Cisco IOS XE Release 3.3S	<p>In Cisco IOS XE Release 3.3S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</p> <p>The following commands were introduced or modified: <b>connect</b> (L2VPN local switching), <b>show connection</b>.</p>
Layer 2 Local Switching - Frame Relay to Frame Relay	Cisco IOS XE Release 3.9S	<p>In Cisco IOS XE Release 3.9S, this feature was introduced on the Cisco ISR 4400 Series Routers.</p>

