Layer 2 Tunnel Protocol Version 3

Feature History

Release	Modification
12.0(21)S	Initial data plane support for the Layer 2 Tunnel Protocol Version 3 (L2TPv3) was introduced on the Cisco 7200 series, Cisco 7500 series, Cisco 10720, and Cisco 12000 series platforms.
12.0(23)S	L2TPv3 control plane support was introduced on the Cisco 7200 series, Cisco 7500 series, Cisco 10720, and Cisco 12000 series platforms.

This document describes the Layer 2 Tunnel Protocol Version 3 feature. This document includes the following sections:

Feature Overview

The Layer 2 Tunnel Protocol Version 3 feature expands on Cisco support of the Layer 2 Tunnel Protocol Version 3 (L2TPv3). Previous Cisco IOS Releases contained only limited support of L2TPv3. This feature introduces L2TPv3 control plane support and new commands for configuring both static and dynamic L2TPv3 sessions.

L2TPv3 is an Internet Engineering Task Force (IETF) l2tpext working group draft that provides several enhancements to L2TP for the capability to tunnel any Layer 2 payload over L2TP. Specifically, L2TPv3 defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network using Layer 2 Virtual Private Networks (VPNs).

L2TP has two fundamental parts:

- A control plane responsible for setting up the connection
- A data plane responsible for tunneling Layer 2 frames.

L2TPv3 signaling is responsible for negotiating control plane parameters, session IDs and cookies; for performing authentication; and for exchanging configuration parameters. L2TPv3 is also used to reliably deliver hello messages and circuit status messages. These messages are critical to support circuit interworking, such as the Local Management Interface (LMI), and to monitor the remote circuit status.

For more information about L2TP, refer to http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/l2tpt.htm and http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/l2tun_ds.htm.

Migration from UTI to L2TPv3

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

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

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

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

L2TPv3 Operation

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

- Xconnect for Layer 2 tunneling via a pseudowire over an IP network
- Layer 2 VPNs for provider edge-to-provider edge (PE-to-PE) router service via Xconnect that support Ethernet, 802.1q (VLAN), Frame Relay, high-level data link control (HDLC) and PPP Layer 2 circuits, including both static (UTI-like) and dynamic (using the new L2TPv3 signaling) forwarded sessions

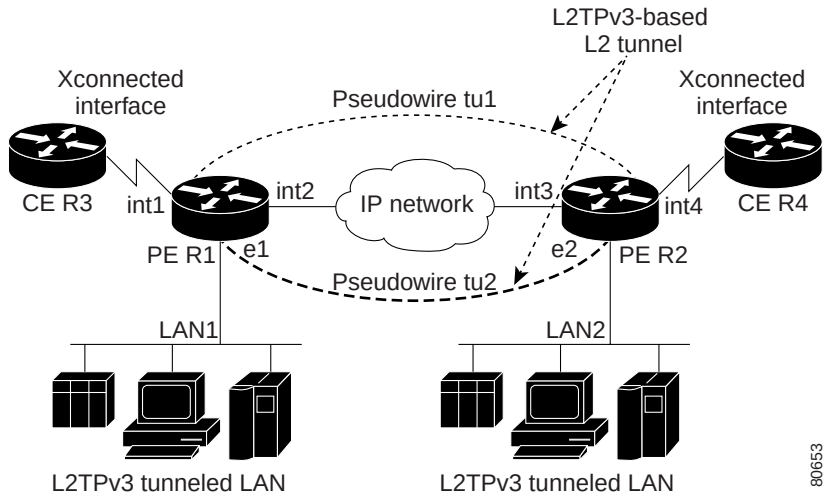
The initial Cisco IOS Release 12.0(23)S features support only the following:

- Layer 2 tunneling (as used in an L2TP access concentrator, or LAC) to an attachment circuit, not Layer 3 tunneling
- L2TPv3 data encapsulation directly over IP (IP protocol number 115), not using User Datagram Protocol (UDP)
- Point-to-point sessions, not point-to-multipoint or multipoint-to-point sessions
- Sessions between the same Layer 2 protocols; for example, Ethernet-to-Ethernet, VLAN-to-VLAN, but not VLAN-to-Ethernet or FrameRelay

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

Figure 1 shows an example of how the L2TPv3 feature is used for setting up VPNs using Layer 2 tunneling over an IP network. All traffic between two customer network sites is encapsulated in IP packets carrying L2TP data messages and sent across an IP network. The backbone routers of the IP network treat the traffic as any other IP traffic and need not know anything about the customer networks.

Figure 1 L2TPv3 Operation



In Figure 1, the PE routers R1 and R2 provide L2TPv3 services. The R1 and R2 routers communicate with each other using a pseudowire over the IP backbone network through a path comprising the interfaces int1 and int2, the IP network, and interfaces int3 and int4.

In this example, the CE routers R3 and R4 communicate through a pair of Xconnect Ethernet or 802.1q 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 via 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.

Please note the following features regarding L2TPv3 operation:

- All packets received on interface int1 will be 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 will be encapsulated directly in IP and sent via the pseudowire session tu2 to R2 interface e2, where it will be sent on LAN2.
- A VLAN on an Ethernet interface can be mapped to an L2TPv3 session.

- For Cisco 12000 series Internet Routers, the other LAN ports on the 8-port Fast Ethernet line card that are *not* being used for L2TPv3 must have a router connected to them: When content-addressable memory (CAM) assisted MAC filtering is turned OFF to allow L2TPv3 to work, it is turned OFF on all ports.

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 Figure 2.

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 via the CLI. Refer to the section "Configuration Tasks" for more information on the CLI commands for L2TPv3.

Figure 2 L2TPv3 Header Format

IP Delivery Header (20 bytes) Protocol ID: 115
L2TPV3 Header consisting of: Session ID (4 bytes) Cookie (0, 4, or 8 bytes) Pseudowire Control Encapsulation (4 bytes by default)
Layer 2 Payload

Session ID

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

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

Session Cookie

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

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

Pseudowire Control Encapsulation

The L2TPv3 pseudowire control encapsulation consists of 32 bits (4 bytes) and contains information used to sequence L2TP packets (see the section "Sequencing"). For the purposes of sequencing, only the first bit and bits 8 to 31 are relevant.

Bit 1 indicates whether the Sequence Number field, bits 8 to 31, contains a valid sequence number and is to be updated.

L2TPv3 Features

L2TPv3 provides Xconnect support for Ethernet, 802.1q (VLAN), Frame Relay, HDLC, and PPP, using the following sessions:

- Static L2TPv3 Sessions (nonnegotiated, PVC-like forwarded sessions)
- Dynamic L2TPv3 Sessions (negotiated, forwarded sessions using the L2TPv3 control plane for session negotiation)

L2TPv3 also includes support for the following features:

- Sequencing
- Local Switching
- Distributed Switching
- L2TPv3 Type of Service Markingg
- Keepalive
- MTU Handling

Static L2TPv3 Sessions

Typically, the L2TP control plane is responsible for negotiating session parameters, such as the session ID or the cookie, in order 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, therefore, set up static L2TPv3 sessions for a PE router by configuring fixed values for the fields in the L2TP data header. A static L2TPv3 session allows the PE to tunnel Layer 2 traffic as soon as the attachment circuit to which the session is bound comes up.

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.

When 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 pairs (AVPs). Each AVP contains information about the nature of the Layer 2 link being forwarded: the payload type, virtual circuit (VC) ID, and so on.

Multiple L2TP sessions (one for each forwarded Layer 2 circuit) can exist between a pair of PEs, 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 SLI message.

Sequencing

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

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

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

Local Switching

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

See the section “Configuration Examples” for an example of how to configure local port switching.

Distributed Switching

Distributed Cisco Express Forwarding (dCEF) switching is supported for L2TP on the Cisco 7500 series and Cisco 12000 series Internet routers, but not on the Cisco 7200 series or Cisco 10720 Internet router.

Note For the Cisco 7500 series, sequencing is supported, but all L2TP packets that require sequence number processing are sent to the Route Switch Processor (RSP). Sequencing is not supported for the Cisco 12000 series Internet routers in Release 12.0(23)S. On th e Cisco 12000 series Internet routers, equencing will be supported in a future release with sequence number processing done by the server card fast path.

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 encapsulate IP packets themselves, 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”.
- Static ToS byte configuration. You specify the ToS byte value used by all packets sent across the pseudowire.

- On the Cisco 10720, ToS configuration can be done using modular quality of service (QoS) command line interface (MQC). If both static ToS byte configuration and MQC ToS byte configuration are implemented, the MQC configuration will take precedence.

See the section "Configuring a Negotiated L2TPv3 Session for Local HDLC Switching" for more information about how to configure ToS information.

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 manually configure sessions.

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), start control channel replay (SCCRP), and start control channel connected (SCCCN) control messages. The control channel is responsible only for maintaining 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 has gone down through the keepalive mechanism, it sends a StopCCN control message and then notifies all of 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 a 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. The one exception is on Cisco 12000 series Internet routers, where fragmentation of tunneled packets is not allowed.
- 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 discovery. When you enable this feature, the following processing is performed:
  - Internet Control Message Protocol (ICMP) unreachable messages sent back to the L2TPv3 router are deciphered and the tunnel MTU is updated accordingly. In order to receive ICMP unreachable messages for fragmentation errors, the Don't Fragment (DF) bit in the tunnel header is set according to the DF bit value received from the CE, or 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. For the packets that match this check, an ICMP unreachable message is sent back to the src IP address of the packet. A Layer 2 header calculation is performed before the ICMP unreachable message is sent to the CE.

L2TPv3 and UTI Feature Comparison

Table 1 compares L2TPv3 and UTI support.

Table 1 Comparison of L2TPv3 and UTI Support

Feature	L2TPv3	UTI
Maximum number of sessions	Cisco 7200 series:3000 Cisco 7500 series: 3000 Cisco 10720: 2000 Cisco 12000 series: 2000	Cisco 7200 series: 1000 Cisco 7500 series: 1000 Cisco 10720 series: 1000 Cisco 12000 series: 1000
Tunnel cookie length	0-, 4-, or 8-byte cookies are supported for the Cisco 7200 series and the Cisco 7500 series routers. For the Cisco 10720 and Cisco 12000 series Internet routers, only 8-byte cookies can be received in Release 12.0(23)S; 0-, 4-, or 8-byte cookies can be sent.	8 bytes
Static sessions	Supported in Release 12.0(21)S.	Supported
Dynamic sessions	Supported in Release 12.0(23)S.	Not supported
Static ToS	Supported in Release 12.0(23)S.	Supported
MQC ToS	Supported in Release 12.0(23)S for the Cisco 10720 only.	Supported
Inner IP ToS mapping	Supported on the Cisco 7200 and Cisco 7500 series routers. To be supported in a future release for the Cisco 10720 and Cisco 12000 series Internet routers.	Not supported
802.1p mapping	Supported in Release 12.0(23)S for the Cisco 10720 only.	Not supported

Feature	L2TPv3	UTI
Keepalive	Supported in Release 12.0(23)S.	Supported on the Cisco 10720 only.
Path MTU discovery	Supported on the Cisco 7200 series, Cisco 7500 series and Cisco 12000 Internet Routers. To be supported in a future release for the Cisco 10720 Internet Router.	Not supported
ICMP unreachable	Supported on the Cisco 7200 series, Cisco 7500 series, and Cisco 12000 Internet routers. To be supported in a future release for the Cisco 10720 Internet router.	Not supported
VLAN rewrite	Supported on the Cisco 7200 series, Cisco 7500 series, and the Cisco 10720 Internet router in Release 12.0(23)S. To be supported in a future release for Cisco 12000 series Internet routers.	Supported
VLAN and non-VLAN translation	To be supported in a future release.	Supported on the Cisco 10720 only.
Port trunking	Supported in Release 12.0(23)S.	Supported
IS-IS packet fragmentation through an L2TPv3 session	Supported on the Cisco 12000 series Internet routers in Release 12.0(23)S. To be supported in a future release for the Cisco 7200 series, Cisco 7500 series, and the Cisco 10720 Internet router.	Not supported
IP packet fragmentation through an L2TPv3 session	To be supported in a future release.	Not supported
Payload sequence number checking	To be supported in a future release.	Not supported
MIB support	VPDN MIB for the pseudowire IfTable MIB for the attachment circuit.	IfTable MIB for the session interface.

Supported L2TPv3 Payloads

L2TPv3 supports the following Layer 2 payloads that can be included in L2TPv3 packets tunneled over the pseudowire:

- Frame Relay
- Ethernet
- 802.1q (VLAN)
- High-Level Data Link Control
- PPP

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

Frame Relay

L2TPv3 supports the Frame Relay functionality described in the following sections:

- Port-to-Port Trunking
- DLCI-to-DLCI Switching
- PVC Status Signaling
- Sequencing
- ToS Marking
- CIR Guarantees

Port-to-Port Trunking

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

For example, in Figure 1, if the two CE routers are connected by a virtual leased line, the PE routers transparently transport all packets between CE R3 and CE R4 over a pseudowire. PE R1 and PE R2 do examine or change the data-link connection identifiers (DLCIs), and do not participate in the LMI protocol. The two CE routers are LMI peers. There is nothing Frame Relay-specific about this service as far as the PE routers are concerned. The CE routers should be able to use any encapsulation based on HDLC framing without needing to change the provider configuration.

DLCI-to-DLCI Switching

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

For example, in Figure 1, CE R3 and PE R1 are Frame Relay LMI peers; CE R4 and PE R2 are also LMI peers. You can use a different type of LMI between CE R3 and PE R1 compared to what you use between CE R4 and PE R2.

The CE devices may be a Frame Relay switch or end-user device. Each Frame Relay PVC is composed of multiple segments. The DLCI value is local to each segment and is changed as traffic is switched from segment to segment. Note that, in Figure 1, two Frame Relay PVC segments are connected by a pseudowire. Frame Relay header flags (FECN, BECN, C/R, DE) are preserved across the pseudowire.

PVC Status Signaling

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

- UNI DTE mode—PVC status is not reported, only received.

- UNI DCE mode—PVC status is reported but not received.
- NNI mode—PVC status is reported and received independently.

L2TPv3 supports all three modes.

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

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

For example, in Figure 1, CE R3 \ reports ACTIVE to PE R1 only if the PVC is available within CE R3. When CE R3 is a switch, it reports all the way to the user device in the customer network.

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

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

Sequencing

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

ToS Marking

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

CIR Guarantees

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

Note In Cisco IOS Release 12.0(23)S, CIR guarantees are supported only on the Cisco 7500 series with dCEF. This support requires that the core has sufficient bandwidth to handle all CE traffic and that the congestion occurs only at the egress PE.

Ethernet

An Ethernet frame arriving at a PE router 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 the interface.

Note Due to the way in which L2TPv3 handles Ethernet frames, an Ethernet interface must be configured to promiscuous mode in order to capture all traffic received on the Ethernet segment attached to the router. All frames will be tunneled through the L2TP pseudowire.

802.1q (VLAN)

L2TPv3 supports VLAN membership 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 Due to the way in which L2TPv3 handles 802.1q VLAN packets, the Ethernet interface must be configured in promiscuous mode to capture all traffic received on the Ethernet segment attached to the router. All frames are tunneled through the L2TP pseudowire.

High-Level Data Link Control

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

PPP

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

Benefits

L2TPv3 Simplifies Deployment of VPNs

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

L2TPv3 Does Not Require MPLS

With L2TPv3 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 higher revenue.

L2TPv3 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.

Restrictions

The following subsections contain information on restrictions:

- Supported Port Adapters for the Cisco 7200 and 7500 Series Routers
- Supported Line Cards for the Cisco 10720 Internet Router
- Supported Line Cards for the Cisco 12000 Series Internet Routers
- General L2TPv3 Restrictions
- Cisco 7500-Specific Restrictions
- Cisco 10720-Specific Restrictions
- Cisco 12000 Series-Specific Restrictions
- Frame Relay-Specific Restrictions
- VLAN-Specific Restrictions

Supported Port Adapters for the Cisco 7200 and 7500 Series Routers

L2TPv3 is supported on the following port adapters in the Cisco 7200 and 7500 series routers:

- Single-port Fast Ethernet 100BASE-TX
- Single-port Fast Ethernet 100BASE-FX
- Dual-port Fast Ethernet 100BASE-TX
- Dual-port Fast Ethernet 100BASE-FX
- Gigabit Ethernet port adapter
- 12-port Ethernet/2-port FE adapter
- 4-port synchronous serial port adapter
- Enhanced 4-port synchronous serial port adapter
- 8-port synchronous serial port adapter
- Single-port HSSI adapter
- Dual-port HSSI adapter

- 8-port multichannel E1 G.703/G.704 120-ohm interfaces
- 2-port multichannel E1 G.703/G.704 120-ohm interfaces
- 8-port multichannel T1 with integrated DSUs
- 8-port multichannel T1 with integrated CSUs and DSUs
- 4-port multichannel T1 with integrated CSUs and DSUs
- 2-port multichannel T1 with integrated CSUs and DSUs
- 8-port multichannel T1/E1
- 1-port multichannel T3 interface
- 1-port multichannel E3 interface
- 2-port enhanced multichannel T3 port adapter
- Single-port T3 port adapter
- Single-port E3 port adapter
- 2-port T3 port adapter
- 2-port T3 port adapter
- Single-port PoS, single-mode, long reach
- Single-port PoS, single-mode, intermediate reach
- Single-port PoS, multimode

L2TPv3 is supported on the following port adapters for the Cisco 7200 series routers only:

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

Supported Line Cards for the Cisco 10720 Internet Router

L2TPv3 is supported on the following uplink and access cards in the Cisco 10720 Internet router:

- 24-port 10/100-Mbps Ethernet TX access card
- 24-port 100-Mbps Ethernet FX access card—multimode (MM)
- 24-port 100-Mbps Ethernet FX access card—single-mode (SM)
- Gigabit Ethernet SFP module—short reach line card
- Gigabit Ethernet SFP module—intermediate reach line card

Supported Line Cards for the Cisco 12000 Series Internet Routers

Table 1 shows the line cards that support L2TPv3 for the Cisco 12000 series Internet routers.

Supported line cards for the Cisco 12000 series Internet routers

Line Card	Engine Type	Port Session	DLCI Session	VLAN Session
PoS	Engine 0	Supported	Supported	Unsupported
PoS	Engine 2	Supported	Supported	Unsupported
2-port Ch OC-3/STM-1(DS1/E1)	Engine 0	Supported	Supported	Unsupported
6-port Ch T3	Engine 0	Supported	Supported	Unsupported
3-port Gigabit Ethernet	Engine 2	Supported	Unsupported	Supported
1-port Gigabit Ethernet	Engine 1	Supported	Unsupported	Supported

Line Card	Engine Type	Port Session	DLCI Session	VLAN Session
8-port Fast Ethernet	Engine 1	Supported	Unsupported	Supported
6-port E3	Engine 0	Supported	Supported	Unsupported
12-port E3	Engine 0	Supported	Supported	Unsupported
6-port DS3	Engine 0	Supported	Supported	Unsupported
12-port DS3	Engine 0	Supported	Supported	Unsupported

General L2TPv3 Restrictions

- CEF must be enabled for the L2TPv3 feature to function. The Xconnect configuration submode is blocked until CEF is enabled. On distributed platforms, such as the Cisco 7500 and Cisco 12000 series, if CEF is disabled while a session is established, the session is torn down and remains down until CEF is reenabled. To enable CEF, use the **ip cef** or **ip cef distributed** command.
- The IP local interface must be a loopback interface. Configuring any other interface with the **ip local interface** command will result in a nonoperational setting.
- The number of sessions on a PPP, HDLC, Ethernet, or 802.1q VLAN port 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 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 sessions, which exceeds the IDB limitations for any Cisco platform in Release 12.0S.
- 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 LMI.
- Static L2TPv3 sessions do not support Frame Relay LMI interworking.
- Static L2TPv3 sessions do not interoperate with UTI using keepalives.
- The **ip pmtu** command used to configure the pseudowire class (see the section "Configuring the L2TPv3 Pseudowire") is not supported for static L2TPv3 sessions. As a result, IS-IS fragmentation through a static L2TPv3 session is not supported.

Cisco 7500-Specific Restrictions

- Although L2TPv3 sequencing is supported on Cisco 7500 series routers, all L2TP packets that require sequence number processing will be sent to the Route Switch Processor (RSP) module.

Cisco 10720-Specific Restrictions

- L2TPv3 sequencing is not supported.
- Although the Cisco 10720 Internet router can send cookie lengths of 0-, 4-, or 8-bytes to an L2TP peer device, it supports the reception of only 8-byte L2TPv3 cookies. Support for reception of 0- and 4-byte cookie lengths will be introduced in future releases.
- The ToS reflect feature is not supported.
- The reassembly of L2TPv3 packets is not supported.
- The Path MTU feature is not supported.

- On the Cisco 10720 Internet router, the **uti translation** command is not migrated for Xconnect service and is not supported in Cisco IOS Release 12.0(23)S. Although the **uti** command is supported in L2TPv3 releases, the **translation** option is lost in the migration.

Cisco 12000 Series-Specific Restrictions

- The variable cookie size, VLAN rewrite, sequencing, ip tos reflect, ip dfbit, fragmentation, and the counters for out-of-order and exceeded session MTU packets are not supported in Release 12.0(23)S.
- IS-IS protocol packet fragmentation is only supported for dynamic L2TPv3 sessions.
- The IP local interface must be a local loopback interface. Configuring any other interface as the IP local interface will result in non-operational sessions.
- The IP local interface must be dedicated for the use of L2TPv3 sessions. This interface must not be shared by any other routing or tunneling protocols.
- Hairpinning is not supported for local-to-local switching. The start and end of an L2TPv3 session must terminate on different routers linked via an IP or MPLS backbone.
- The aggregate performance is bound by the server card limit of 2.5 million packets per second (pps).
- The dedicated tunnel server card 1-Port OC-48c/STM-16c POS/SDH is required for L2TPv3 to function. The server card will not run any Engine 2 features.
- The **ip unnumbered** command and IP address should be configured under the PoS interface of the server card prior to hardware-module configuration. This configuration makes the server card IP-Aware for backbones requiring an Address Resolution Protocol (ARP) to be generated by the line card. The backbone types that require this configuration are Ethernet and spatial reuse protocol (SRP). This configuration is also a requirement for session keepalives. The interface port of the server card will automatically be set to loopback internal and no keepalives once the **hw-module slot** *slot-number* **mode server** command is configured.
- Due to a framer problem, the server card interfaces accounting in (packets out) will not be accurate.
- Only features found in the Vanilla uCode bundle are supported on Engine 2 line cards that are associated with a L2TPv3 session and on a different interface, DLCI or VLAN of the same line card.
- Configuring Engine 2 features not found in the Vanilla uCode bundle on any port of the Engine 2 line card that has a L2TPv3 session bound to one or more interfaces will cause the Vanilla uCode to be swapped out. This configuration will cause all traffic through the L2TPv3 session to stop on that Engine 2 line card. In this case, rebinding of the L2TPv3 session will be required when the Vanilla uCode bundle is restored.
- Configuring output access control lists (ACLs) on any line card will swap out the running Engine 2 line card Vanilla uCode bundle in favor of the ACL uCode bundle. This configuration will cause all traffic through the L2TPv3 session to stop on those Engine 2 line cards. If output ACLs are essential on the router, it is advisable to originate all L2TPv3 sessions on Engine 0 line cards. Output ACLs will not swap out the server card uCode bundle due to the higher priority.
- Engine 2 line cards do not support Frame Relay switching and Frame Relay L2TPv3 DLCI session on the same line card.
- On Engine 2 line cards, the input Frame Relay PVC counters will not be updated.
- The 8-port Fast Ethernet line card should not be connected to a hub or switch when L2TPv3 is configured on the ingress side of one or more of its ports, or duplicate packets will be generated, causing the router to be flooded with packets. This restriction results from the requirement that CAM filtering is disabled when L2TPv3 is used.

- On the 3-port Gigabyte Ethernet line card, performance degradation can occur if IP packets coming from a port are sent to the slow path for forwarding. This performance degradation will occur if both the following conditions are met:
  - The port has at least one 802.1q subinterface that is in an L2TPv3 session.
  - The IP packet comes from the port interface itself (not 802.1q encapsulated) or from an 802.1q subinterface that is under the port interface but has no L2TPv3 session bound to it.
- On the 1-Port OC-48c/STM-16c POS/SDH linecard, the maximum performance of 2.5 million packets per second is achieved only if you use transmit buffer management (TBM) ASIC ID 60F1. Other ASIC ID versions can cause the performance to be reduced by half. To determine the ASIC value of the line card, use the **execute-on slot** *slot-number* **show controller frfab bma reg | include asic** command, where *slot-number* is the slot number of the server card.
- The optics of the 1-Port OC-48c/STM-16c POS/SDH line card should be covered due to possible interference or noise causing CRC errors on the line card. These errors are caused by a framer problem in the line card.

Frame Relay-Specific Restrictions

- Frame Relay per-DLCI forwarding and port-to-port trunking are mutually exclusive. L2TPv3 does not support the use of both on the same interface at the same time.
- The **xconnect** command is not supported on Frame Relay interfaces directly. For Frame Relay, the Xconnect is applied under the **connect** command specifying the DLCI to be used.
- Changing the encapsulation type on any interface removes any existing **xconnect** command applied to that interface.
- The DE bit value does not influence the ToS bits.
- To use DCE or a Network-to-Network Interface (NNI) on a Frame Relay port, you must configure the **frame-relay switching** command.
- The MQC supported by L2TPv3 on Frame Relay interfaces is supported only on the Cisco 7500 series with dCEF and is limited to using the **match fr-dlci** command with the **bandwidth** command.
- Frame Relay policing is nondistributed on the Cisco 7500 series. By configuring Frame Relay policing, you cause traffic on the affected PVCs to be sent to the RSP for processing. Frame Relay policing is not supported on the Cisco 12000 series Internet router.
- Frame Relay support is for 10-bit DLCI addresses. Frame Relay Extended Addressing is not supported.
- Multi-point DLCI is not supported.
- The keepalive will automatically be disabled on interfaces that have an Xconnect applied to them, except for Frame Relay encapsulation which is a requirement for LMI.
- Static L2TPv3 sessions will not support Frame Relay LMI interworking.

VLAN-Specific Restrictions

- A PE router is responsible only for static VLAN membership entries that are manually configured on the router. Dynamic VLAN membership entries, entry aging, and membership discovery are not supported.
- Implicit tagging for VLAN membership operating on the other layers (such as at Layer 2, membership by MAC address or protocol type, at Layer 3, or membership by IP subnet) 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.
- VLAN ID rewrite is not supported on the Cisco 12000 series Internet router.

Related Features and Technologies

- Wide-area networking
- VPNs
- L2TP
- UTI

Related Documents

- *Layer 2 Tunneling Protocol Version 3 Technical Overview*
- *Cisco IOS Release 12.0 Configuration Fundamentals Configuration Guide*
- *Cisco IOS Release 12.0 Configuration Fundamentals Command Reference*
- *Cisco IOS Release 12.0 Dial Solutions Command Reference*
- *Network Protocols Command Reference, Part 1*, Release 11.3
- *Cisco IOS Release 12.0 Wide-Area Networking Command Reference*
- *Universal Transport Interface (UTI)*
- *Layer 2 Tunnel Protocol*
- *Layer 2 Tunneling Protocol: A Feature in Cisco IOS Software*

Supported Platforms

- Cisco 7200 series
- Cisco 7500 series
- Cisco 10720
- Cisco 12000 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- draft-ietf-l2tpext-l2tp-base-03.txt

MIBs

- VPDN MIB—MIB support for L2TPv3 is based on the VPDN MIB.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- RFC 2661 (L2TP)

Prerequisites

- Before you configure an Xconnect attachment circuit for a CE device (see the section "Configuring the Xconnect Attachment Circuit"), the CEF feature must be enabled. To enable CEF 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.
- To enable SNMP notifications of L2TP session up and down events, enter the **snmp-server enable traps l2tun session** command before configuring L2TPv3.

Configuration Tasks

See the following sections for configuration tasks for the L2TPv3 feature. Each task in the list is identified as either required or optional.

- Configuring L2TP Control Channel Parameters (optional)
- Configuring the L2TPv3 Pseudowire (required)
- Configuring the Xconnect Attachment Circuit (required)
- Manually Configuring L2TPv3 Session Parameters (required)
- Verifying an L2TPv3 Session (optional)
- Verifying an L2TP Control Channel (optional)

Configuring L2TP 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 router at each end of the control channel. Configuring L2TP control channel parameters is optional. However, the L2TP class must be configured before it is with associated a pseudowire class (see the section "Configuring the L2TPv3 Pseudowire").

The three main groups of L2TP control channel parameters that you can configure in an L2TP class are described in the following sections:

- Configuring L2TP Control Channel Timing Parameters
- Configuring L2TP Control Channel Authentication Parameters
- Configuring L2TP Control Channel Maintenance 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 L2TP class control channel 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

To configure a set of timing control channel parameters in an L2TP class, use the following commands beginning in global configuration mode. All of the timing control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, the default values will be applied.

	Command	Purpose
Step 1	`Router(config)# l2tp-class [l2tp-class-name]`	Specifies the L2TP class name and enters L2TP class configuration mode. The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name* for each one.
Step 2	`Router(config-l2tp-class)# receive-window size`	(Optional) Configures the number of packets that can be received by the remote peer before backoff queuing occurs. The valid values range from 1 to the upper limit the peer has for receiving packets. The default value is the upper limit.
Step 3	`Router(config-l2tp-class)# retransmit {initial retries initial-retries| retries retries| timeout {max | min} timeout}`	(Optional) Configures parameters that affect the retransmission of control packets: • **initial retries**—specifies how many SCCRQs are resent before giving up on the session. Valid values for the *initial-retries* argument range from 1 to 1000. The default value is 2. • **retries**—specifies how many retransmission cycles occur before determining that the peer PE router does not respond. Valid values for the *retries* argument range from 1 to 1000. The default value is 15. • **timeout {max | min}**—specifies maximum and minimum retransmission intervals (in seconds) for resending control packets. Valid values for the *timeout* argument range from 1 to 8. The default maximum interval is 8; the default minimum interval is 1.
Step 4	`Router(config-l2tp-class)# timeout setup seconds`	(Optional) Configures the amount of time, in seconds, allowed to set up a control channel. Valid values for the *seconds* argument range from 60 to 6000. The default value is 300.

Configuring L2TP Control Channel Authentication Parameters

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

- Authentication for the L2TP control channel
- Local host name used for authenticating the control channel

- Hiding the AVPs in outgoing control messages
- Password used for control channel authentication and AVP hiding

To configure a set of authentication control channel parameters in an L2TP class, use the following commands beginning in global configuration mode. All of the authentication control channel parameter configurations are optional and may be configured in any order. If these parameters are not configured, the default values will be applied.

	Command	Purpose
Step 1	`Router(config)# l2tp-class [l2tp-class-name]`	Specifies the L2TP class name and enters L2TP class configuration mode. The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name* for each one.
Step 2	`Router(config-l2tp-class)# authentication`	(Optional) Enables authentication for the control channel between PE routers.
Step 3	`Router(config-l2tp-class)# hostname name`	(Optional) Specifies a host name used to identify the router during L2TP control channel authentication. If you do not use this command, the default host name of the router is used.
Step 4	`Router(config-l2tp-class)# hidden`	(Optional) Hides the AVPs in control messages. AVP hiding is not hidden by default.
Step 5	`Router(config-l2tp-class)# password [encryption-type] password`	(Optional) Configures the password used for control channel authentication. The valid values for the optional encryption type range from 0 to 7. If you do not use this command to specify a password, the password associated with the remote peer PE is taken from the value entered with the **username password** *value* global configuration command.

Configuring L2TP Control Channel Maintenance Parameters

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

To configure the interval used for hello messages in an L2TP class, use the following commands beginning in global configuration mode. This control channel parameter configuration is optional. If this parameter is not configured, the default value will be applied.

	Command	Purpose
Step 1	`Router(config)# l2tp-class [l2tp-class-name]`	Specifies the L2TP class name and enters L2TP class configuration mode. The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name* for each one.
Step 2	`Router(config-l2tp-class)# hello interval`	(Optional) Specifies the exchange interval (in seconds) used between L2TP hello packets. Valid values for the *interval* argument range from 0 to 1000. The default value is 60.

Configuring the L2TPv3 Pseudowire

The pseudowire class configuration procedure creates a configuration template for the pseudowire. You use this template, or class, to configure session-level parameters for L2TPv3 sessions that will be 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, payload-specific options, and IP properties. The setting that determines if signaling is used to set up the pseudowire is also included.

For simple L2TPv3 signaling configurations on most platforms, pseudowire class configuration is optional. However, specifying a source IP address to configure a loopback interface is highly recommended. If you do not configure a loopback interface, the router will choose the best available local address, which could be any IP address configured on a core-facing interface. This configuration could potentially prevent a control channel from being established. On the Cisco 12000 series Internet Routers, specifying a source IP address is mandatory, and should configure a loopback interface that is dedicated for the use of L2TPv3 sessions exclusively. If you do not configure other pseudowire class configuration commands, the default values are used.

Once you specify the **encapsulation l2tpv3** command, you cannot remove it using the **no encapsulation l2tpv3** command. Nor can you change the command's setting using the **encapsulation mpls** command. Those 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 with the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command and re-establish the pseudowire and specify the new encapsulation type.

To configure a pseudowire class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router# pseudowire-class [pw-class-name]`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
Step 2	`Router(config-pw)# encapsulation l2tpv3`	Specifies that L2TPv3 is used as the data encapsulation method to tunnel IP traffic.
Step 3	`Router(config-pw)# protocol {l2tpv3 | none} [l2tp-class-name]`	(Optional) Specifies the L2TPv3 signaling protocol to be used to manage the pseudowires created with the control channel parameters in the specified L2TP class (see the section "Configuring L2TP Control Channel Parameters"). If the *l2tp-class-name* argument is not specified, the default values for L2TP control channel parameters will be used. The default **protocol** option is **l2tpv3**. If you do not want to use signaling in the L2TPv3 sessions created with this pseudowire class, enter **protocol none**. (The **protocol none** configuration is necessary when configuring interoperability with a remote peer that runs UTI.)
Step 4	`Router(config-pw)# ip local interface interface-name`	Specifies the PE router interface whose IP address is to be used as the source IP address for sending tunneled packets. Use the same local interface name for all pseudowire classes configured between a pair of PE routers.

	Command	Purpose
Step 5	`Router(config-pw)# ip pmtu`	(Optional) Enables the discovery of the path maximum transmission unit (MTU) for tunneled traffic. This command enables the processing of ICMP unreachable messages that indicate fragmentation errors in the backbone network that carries L2TPv3 session traffic. Also, this command enables MTU checking for IP packets sent into the session and that have the DF bit set. Any IP packet larger than the MTU is dropped and an ICMP unreachable message is sent. MTU discovery is disabled by default. **Note** The **ip pmtu** command is not supported if you disabled signaling with the **protocol none** command in Step 3.
Step 6	`Router(config-pw)# ip tos {value` *value* `| reflect}`	(Optional) Configures the value of the ToS byte in IP headers of tunneled packets, or reflects the ToS byte value from the inner IP header. Valid values for the *value* argument range from 0 to 255. The default ToS byte value is 0.
Step 7	`Router(config-pw)# ip dfbit set`	(Optional) Configures the value of the DF bit in the outer headers of tunneled packets. Use this command if (for performance reasons) you do not want reassembly of tunneled packets to be performed on the peer PE router. This command is disabled by default.
Step 8	`Router(config-pw)# ip ttl` *value*	(Optional) Configures the value of the time to live (TTL) byte in the IP headers of tunneled packets. Valid values for the *value* argument range from 1 to 255. The default TTL byte value is 255.
Step 9	`Router(config-pw)# ip protocol {l2tp | uti |` *protocol-number*`}`	(Optional) Configures the IP protocol to be used for tunneling packets. For backward compatibility with UTI, enter **uti** or **120**, the UTI protocol number. The default IP protocol value is **l2tp** or **115**, the L2TP protocol number.
Step 10	`Router(config-pw)# sequencing {transmit | receive | both}`	(Optional) Specifies the direction in which sequencing of data packets in a pseudowire is enabled: • **transmit**—Updates the Sequence Number field in the headers of data packets sent over the pseudowire according to the data encapsulation method that is used. • **receive**—Keeps the Sequence Number field in the headers of data packets received over the pseudowire. Out-of-order packets are dropped. • **both**—Enables both the **transmit** and **receive** options.

Configuring the Xconnect Attachment Circuit

This configuration procedure binds an Ethernet, 802.1q VLAN, or Frame Relay attachment circuit to an L2TPv3 pseudowire for Xconnect service. The virtual circuit identifier that you configure creates the binding between a pseudowire configured on a PE router and an attachment circuit in a CE device. The virtual circuit identifier configured on the PE router at one end of the L2TPv3 control channel must also be configured on the peer PE router at the other end.

To configure an Xconnect attachment circuit, use the following command beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# xconnect` *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]	Specifies the IP address of the peer PE router and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel. The peer router ID (IP address) and virtual circuit ID must be a unique combination on the router. At least one of the following pseudowire class parameters must be configured for the *pseudowire-parameters* argument: • **encapsulation** {**l2tpv3** [**manual**] | **mpls**}—Specifies the tunneling method used to encapsulate data in the pseudowire: – **l2tpv3**—L2TPv3 is the tunneling method to be used. – **manual**—(Optional) No signaling is to be used in the L2TPv3 control channel. This command places the router in xconnect configuration mode for manual configuration of L2TPv3 parameters for the attachment circuit. – **mpls**—Multiprotocol Label Switching (MPLS) is the tunneling method to be used. • **pw-class** {*pw-class-name*}—The pseudowire class configuration from which the data encapsulation type (L2TPv3) will be taken. The optional **encapsulation** parameter specifies the method of pseudowire tunneling used: L2TPv3 or MPLS. Enter **manual** if you do not want signaling used in the L2TPv3 control channel. The **encapsulation l2tpv3 manual** keyword combination enters xconnect configuration submode. See the section "Manually Configuring L2TPv3 Session Parameters" for the other L2TPv3 commands that you must enter to complete the configuration of the L2TPv3 control channel. If you do not enter an **encapsulation** value, the encapsulation method entered with the **password** command in the section "Configuring the Xconnect Attachment Circuit" is used. The optional **pw-class** parameter binds the Xconnect statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. Specify the pseudowire-class option if you need to configure more advanced options. **Note** You must configure either the **encapsulation** or the **pw-class** option. You may configure both options. The optional **sequencing** parameter specifies whether or not sequencing is required for packets that are received, sent, or both received and sent.

Manually Configuring L2TPv3 Session Parameters

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

To manually configure L2TPv3 control channel parameters on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Specifies the interface by type (for example, Ethernet) and slot and port number, and enters interface configuration mode.
Step 2	`Router(config-if)# xconnect peer-router-id vc-id encapsulation l2tpv3 manual`	Specifies the IP address of the peer PE router and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel. The peer router ID (IP address) and virtual circuit identifier must be a unique combination on the router. The **encapsulation l2tpv3 manual** parameter specifies that L2TPv3 is to be used as the pseudowire tunneling method, and enters xconnect configuration mode.
Step 3	`Router(config-if-xconn)# l2tp id local-session-id remote-session-id`	Configures the identifiers for the local L2TPv3 session and for the remote L2TPv3 session on the peer PE router. This command is required to complete the attachment circuit configuration and for a static L2TPv3 session configuration.
Step 4	`Router(config-if-xconn)# l2tp cookie local size low-value [high-value]`	(Optional) Specifies the value that the peer PE must include in the cookie field of incoming (received) L2TP packets. 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. 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. **Note** This command must be configured on the Cisco 12000 series Internet routers.

	Command	Purpose
Step 5	`Router(config-if-xconn)# l2tp cookie remote size low-value [high-value]`	(Optional) Specifies the value that the router includes in the cookie field of outgoing (sent) L2TP packets. 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. 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. **Note** This command must be configured on the Cisco 12000 series Internet routers.
Step 6	`Router(config-if-xconn)# l2tp hello l2tp-class-name`	(Optional) Specifies the L2TP class name to use (see the section "Configuring L2TP Control Channel Parameters") for control channel configuration parameters, including the interval to use between hello keepalive messages. 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.

Verifying an L2TPv3 Session

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

```
Router# show l2tun session all

Tunnel Information Total tunnels 1 sessions 1

Tunnel id 26515 is up, remote id is 41814, 1 active sessions
  Tunnel state is established, time since change 03:11:50
  Tunnel transport is IP (115)
  Remote tunnel name is tun1
    Internet Address 172.18.184.142, port 0
  Local tunnel name is Router
    Internet Address 172.18.184.116, port 0
  Tunnel domain is
  VPDN group for tunnel is
  0 packets sent, 0 received
  0 bytes sent, 0 received
  Control Ns 11507, Nr 11506
  Local RWS 2048 (default), Remote RWS 800
  Tunnel PMTU checking disabled
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 1, max 1
  Total resends 0, ZLB ACKs sent 11505
  Current nosession queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0 0
  Sessions disconnected due to lack of resources 0
```

Verifying an L2TP Control Channel

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

```
Router# show l2tun tunnel all

Tunnel Information Total tunnels 1 sessions 1

Tunnel id 26515 is up, remote id is 41814, 1 active sessions
  Tunnel state is established, time since change 03:11:50
  Tunnel transport is IP (115)
  Remote tunnel name is tun1
    Internet Address 172.18.184.142, port 0
  Local tunnel name is Router
    Internet Address 172.18.184.116, port 0
  Tunnel domain is
  VPDN group for tunnel is
  0 packets sent, 0 received
  0 bytes sent, 0 received
  Control Ns 11507, Nr 11506
  Local RWS 2048 (default), Remote RWS 800
  Tunnel PMTU checking disabled
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 1, max 1
  Total resends 0, ZLB ACKs sent 11505
  Current nosession queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0 0
  Sessions disconnected due to lack of resources 0
```

Configuration Examples

This section provides the following configuration examples:

- Configuring Frame Relay DLCI-to-DLCI Switching
- Configuring Frame Relay Trunking
- Configuring an MQC for Committed Information Rate Guarantees
- Configuring a Static L2TPv3 Session for an Xconnect Ethernet Interface
- Configuring a Negotiated L2TPv3 Session for an Xconnect VLAN Subinterface
- Configuring a Negotiated L2TPv3 Session for Local HDLC Switching

Configuring Frame Relay DLCI-to-DLCI Switching

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

```
Router(config)# pseudowire-class fr-xconnect
Router(config-pw)# encapsulation l2tpv3
Router(config-pw)# protocol l2tpv3
Router(config-pw)# source interface Loopback0
Router(config-pw)# sequencing both

Router(config)# interface Serial0/0
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay intf-type dce

Router(config)# connect one Serial0/0 100 l2transport
Router(config-if)# xconnect 10.0.3.201 555 pw-class fr-xconnect

Router(config)# connect two Serial0/0 200 l2transport
Router(config-if)# xconnect 10.0.3.201 666 pw-class fr-xconnect
```

Configuring Frame Relay Trunking

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

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

```
Router(config)# interface Serial0/0
Router(config-if)# xconnect 10.0.3.201 555 pw-class serial-xconnect
```

Configuring an MQC for Committed Information Rate Guarantees

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

```
Router(config)# ip cef distributed
Router(config)# class-map dlci100
Router(config-cmap)# match fr-dlci 100
Router(config)# class-map dlci200
Router(config-cmap)# match fr-dlci 200

Router(config)# policy-map dlci
Router(config-pmap)# class dlci100
Router(config-pmap-c)# bandwidth 256
Router(config-pmap)# class dlci200
Router(config-pmap-c)# bandwidth 512

Router(config)# interface Serial0/0
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay intf-type dce
Router(config-if)# service-policy output dlci

Router(config)# connect one Serial0/0 100 l2transport
Router(config-if)# xconnect 10.0.3.201 555 encapsulation l2tpv3

Router(config)# connect two Serial0/0 200 l2transport
Router(config-if)# xconnect 10.0.3.201 666 encapsulation l2tpv3
```

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 router 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.

```
Router(config)# l2tp-class l2tp-defaults
Router(config-l2tp-class)# retransmit initial retries 30
Router(config-l2tp-class)# cookie-size 8

Router(config)# pseudowire-class ether-pw
Router(config-pw)# encapsulation l2tpv3
Router(config-pw) protocol none
Router(config-pw)# ip local interface e0/0

Router(config)# interface Ethernet 0/0
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn)# l2tp id 222 111
Router(config-if-xconn)# l2tp cookie local 4 54321
Router(config-if-xconn)# l2tp cookie remote 4 12345
Router(config-if-xconn)# l2tp hello l2tp-defaults
```

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.

```
Router(config)# l2tp-class class1
Router(config-l2tp-class)# authentication
Router(config-l2tp-class)# password secret

Router(config)# pseudowire-class vlan-xconnect
Router(config-pw)# encapsulation l2tpv3
Router(config-pw)# protocol l2tpv3 class1
Router(config-pw)# ip local interface e0/0

Router(config)# interface Ethernet0/0.1
Router(config-if)# encapsulation dot1Q 5
Router(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect
```

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.

```
Router(config)# interface loopback 1
Router(config-if)# ip address 10.0.0.1 255.255.255.255

Router(config)# interface loopback 2
Router(config-if)# ip address 10.0.0.2 255.255.255.255

Router(config)# pseudowire-class loopback1
Router(config-pw)# encapsulation l2tpv3
Router(config-pw)# ip local interface loopback1

Router(config)# pseudowire-class loopback2
Router(config-pw)# encapsulation l2tpv3
Router(config-pw)# ip local interface loopback2

Router(config)# interface s0/0
Router(config-if)# encapsulation hdlc
Router(config-if)# xconnect 10.0.0.1 100 pw-class loopback2

Router(config)# interface s0/1
Router(config-if)# encapsulation hdlc
Router(config-if)# xconnect 10.0.0.2 100 pw-class loopback1
```

Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

- **authentication**
- **debug acircuit**
- **debug vpdn**
- **debug xconnect**
- **debug acircuit**
- **encapsulation l2tpv3**
- **hello**
- **hidden**
- **hostname**
- **ip dfbit set**
- **ip local interface**
- **ip pmtu**
- **ip protocol**
- **ip tos**
- **ip ttl**
- **l2tp-class**
- **l2tp cookie local**
- **l2tp cookie remote**
- **l2tp hello**
- **l2tp id**
- **password**
- **protocol**
- **pseudowire-class**
- **receive-window**
- **retransmit**
- **sequencing**
- **show l2tun session**
- **show l2tun tunnel**
- **snmp-server enable traps l2tun session**
- **timeout setup**
- **xconnect**

authentication

To enable Layer 2 Tunnel Protocol Version 3 (L2TPv3) authentication, use the **authentication** command in L2TP class configuration mode. To disable L2TPv3 authentication, use the **no** form of this command.

authentication

no authentication

Syntax Description

This command has no arguments or keywords.

Defaults

L2TPv3 authentication is disabled.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the **authentication** command to enable L2TPv3 authentication.

Examples

The following example enables authentication in L2TPv3 pseudowires configured using the L2TP class configuration named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# authentication
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.
password	Configures the password used by a PE router for L2TPv3 authentication.

debug acircuit

To troubleshoot events and failures related to an attachment circuit, use the **debug acircuit** command in privileged EXEC mode. To disable the **debug acircuit** command, use the **no** form of this command.

debug acircuit {**error** | **event**}

no debug acircuit {**error** | **event**}

Syntax Description

error	Displays errors that occur in attachment circuits.
event	Displays events that occur in attachment circuits.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the **debug acircuit** command to identify provisioning events, setup failures, circuit up and down events, and configuration failures.

Examples

The following example shows output from the **debug acircuit** command for an Xconnect session on an Ethernet interface:

```
23:28:35: ACLIB [10.0.3.201, 5]: SW AC interface UP for Ethernet interface Et2/1
23:28:35: ACLIB [10.0.3.201, 5]: pthru_intf_handle_circuit_up() calling acmgr_circuit_up
23:28:35: ACLIB [10.0.3.201, 5]: Setting new AC state to Ac-Connecting
23:28:35: ACLIB [10.0.3.201, 5]: SW AC interface UP for Ethernet interface Et2/1
23:28:35: ACLIB [10.0.3.201, 5]: pthru_intf_handle_circuit_up() ignoring up event. Already
connected or connecting.
23:28:35: ACMGR: Receive <Circuit Up> msg
23:28:35: Et2/1 ACMGR: circuit up event, SIP state chg down to connecting, action is
service request
23:28:35: Et2/1 ACMGR: Sent a sip service request
23:28:37: %LINK-3-UPDOWN: Interface Ethernet2/1, changed state to up
23:28:38: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet2/1, changed state to up
23:28:53: Et2/1 ACMGR: Rcv SIP msg: resp connect forwarded, hdl D6000002, sss_hdl 9E00000F
23:28:53: Et2/1 ACMGR: service connected event, SIP state chg connecting to connected,
action is respond forwarded
23:28:53: ACLIB: pthru_intf_response hdl is D6000002, response is 1
23:28:53: ACLIB [10.0.3.201, 5]: Setting new AC state to Ac-Connected
```

Related Commands

Command	Description
debug vpdn	Displays errors and events relating to L2TP configuration and the surrounding Layer 2 tunneling infrastructure.
debug xconnect	Displays errors and events related to an Xconnect configuration.

debug vpdn

To troubleshoot Layer 2 Tunnel Protocol Version 3 (L2TPv3) and the surrounding Layer 2 tunneling infrastructure, use the **debug vpdn** command in privileged EXEC mode. To disable the **debug vpdn** command, use the **no** form of this command.

debug vpdn {**error** | **event** | **l2x-errors** | **l2x-events** | **l2x-packets** | **packet** | **packet detail** | **packet errors**}

no debug vpdn {**error** | **event** | **l2x-errors** | **l2x-events** | **l2x-packets** | **packet** | **packet detail** | **packet errors**}

Syntax Description

error	Displays errors that occur in protocol-independent conditions.
event	Displays events resulting from protocol-independent conditions.
l2x-errors	Displays errors that occur in protocol-specific conditions.
l2x-events	Displays events resulting from protocol-specific conditions.
l2x-packets	Displays detailed information about control packets in protocol-specific conditions.
packet	Displays information about high-level Layer 2 control packets.
packet detail	Displays detailed packet information, including packet dumps.
packet errors	Displays errors that occur in packet processing.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Note that the **debug vpdn packet** and **debug vpdn packet detail** commands generate several debug operations per packet. Depending on the L2TP traffic pattern, these commands may cause the CPU load to increase to a high level that impacts performance.

Examples

The following example shows output from the **debug vpdn** command for an Xconnect session on an Ethernet interface:

```
23:31:18: L2X: l2tun session [1669204400], event [client request], old state [open], new
state [open]
 23:31:18: L2X: L2TP: Received L2TUN message <Connect>
 23:31:18: Tnl/Sn58458/28568 L2TP: Session state change from idle to wait-for-tunnel
 23:31:18: Tnl/Sn58458/28568 L2TP: Create session
 23:31:18: Tnl58458 L2TP: SM State idle
 23:31:18: Tnl58458 L2TP: O SCCRQ
```

```
 23:31:18: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:18: Tnl58458 L2TP: Tunnel state change from idle to wait-ctl-reply
 23:31:18: Tnl58458 L2TP: SM State wait-ctl-reply
 23:31:18: Tnl58458 L2TP: I SCCRP from router
 23:31:18: Tnl58458 L2TP: Tunnel state change from wait-ctl-reply to established
 23:31:18: Tnl58458 L2TP: O SCCCN to router tnlid 8012
 23:31:18: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:18: Tnl58458 L2TP: SM State established
 23:31:18: Tnl/Sn58458/28568 L2TP: O ICRQ to router 8012/0
 23:31:18: Tnl/Sn58458/28568 L2TP: Session state change from wait-for-tunnel to wait-reply
 23:31:19: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:20: %LINK-3-UPDOWN: Interface Ethernet2/1, changed state to up
 23:31:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet2/1, changed state to
up
 23:31:25: L2X: Sending L2TUN message <Connect OK>
 23:31:25: Tnl/Sn58458/28568 L2TP: O ICCN to router 8012/35149
 23:31:25: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
 23:31:25: Tnl/Sn58458/28568 L2TP: Session state change from wait-reply to established
 23:31:25: L2X: l2tun session [1669204400], event [server response], old state [open], new
state [open]
 23:31:26: Tnl58458 L2TP: Control channel retransmit delay set to 1 seconds
```

Related Commands

Command	Description
debug acircuit	Displays events and failures related to attachment circuits.
debug xconnect	Displays errors and events related to an Xconnect configuration.

debug xconnect

To debug a problem related to the Xconnect configuration, use the **debug xconnect** command in privileged EXEC mode. To disable the **debug xconnect** command, use the **no** form of this command.

debug xconnect {**error** | **event**}

no debug xconnect {**error** | **event**}

Syntax Description

error	Displays errors related to an Xconnect configuration.
event	Displays events related to an Xconnect configuration processing.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use this command to display debugging information about Xconnect sessions.

Examples

The following example shows output from the **debug xconnect** command for a Xconnect session on an Ethernet interface:

```
00:01:16: XC AUTH [Et2/1, 5]: Event: start xconnect authorization, state changed from IDLE
to AUTHORIZING
00:01:16: XC AUTH [Et2/1, 5]: Event: found xconnect authorization, state changed from
AUTHORIZING to DONE
00:01:16: XC AUTH [Et2/1, 5]: Event: free xconnect authorization request, state changed
from DONE to END
```

Related Commands

Command	Description
debug acircuit	Displays events and failures related to attachment circuits.
debug vpdn	Displays errors and events relating to L2TP configuration and the surrounding Layer 2 tunneling infrastructure.

encapsulation l2tpv3

To specify that Layer 2 Tunnel Protocol Version 3 (L2TPv3) is used as the data encapsulation method for tunneling IP traffic over the pseudowire, use the **encapsulation l2tpv3** command in pseudowire class or VC class configuration mode. To remove L2TPv3 as the encapsulation method, use the **no pseudowire-class** command (see the Usage Guidelines for more information).

encapsulation l2tpv3

no pseudowire-class

Syntax Description

This command has no arguments or keywords.

Command Default

No encapsulation method is specified.

Command Modes

Pseudowire class configuration
VC class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(27)SBC	Support for this command was integrated into Cisco IOS Release 12.2(27)SBC.

Usage Guidelines

This command must be configured if the pseudowire class will be referenced from an Xconnect configured to forward L2TPv3 traffic.

Once you specify the **encapsulation l2tpv3** command, you cannot remove it using the **no encapsulation l2tpv3** command. Nor can you change the command's setting using the **encapsulation mpls** command. Those 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 with the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command and re-establish the pseudowire and specify the new encapsulation type.

Examples

The following example shows how to configure L2TPv3 as the data encapsulation method for the pseudowire class named ether-pw:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# encapsulation l2tpv3
```

The following example configures ATM AAL5 over L2TPv3 in VC class configuration mode:

```
vc-class atm aal5class
 encapsulation aal5
```

Related Commands

Command	Description
encapsulation mpls	Configures MPLS as the data encapsulation method over AToM-enabled IP/MPLS networks.
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

hello

To configure the interval used to exchange hello keepalive packets in a Layer 2 Tunnel Protocol Version 3 (L2TPv3) control channel, use the **hello** command in L2TP class configuration mode. To disable the sending of hello keepalive packets, use the **no** form of this command.

hello *interval*

no hello *interval*

Syntax Description

interval	Number of seconds a PE router at one end of an L2TPv3 control channel waits before sending a hello keepalive packet to its peer provider edge (PE) router. The valid values range from 0 to 1000 seconds. The default value is 60 seconds.

Defaults

60 seconds

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

You can configure different values with the **hello** command on the PE router at each end of an L2TPv3 control channel.

Examples

The following example sets an interval of 120 seconds between the sending of hello keepalive messages in L2TPv3 pseudowires configured using the L2TP class configuration named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# hello 120
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

hidden

To hide the attribute-value pair (AVP) values in Layer 2 Tunneling Protocol (L2TP) control messages, use the **hidden** command in L2TP class configuration mode. To unhide AVPs, use the **no** form of this command.

hidden

no hidden

Syntax Description

This command has no arguments or keywords.

Defaults

L2TP AVP hiding is disabled.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the **hidden** command to provide additional security for the exchange of control messages between provider edge routers in a Layer 2 Tunnel Protocol Version 3 (L2TPv3) control channel. Because username and password information is exchanged between devices in clear text, it is useful to encrypt L2TP AVP values with the **hidden** command.

Examples

The following example enables AVP hiding and encrypts AVPs in control messages in L2TPv3 pseudowires configured using the L2TP class configuration named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# hidden
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

hostname

To configure the host name that the router will use to identify itself during Layer 2 Tunnel Protocol Version 3 (L2TPv3) authentication, use the **hostname** command in L2TP class configuration mode. To remove the host name, use the **no** form of this command.

hostname *name*

no hostname *name*

Syntax Description

name	Name used to identify the router during authentication.

Defaults

No host name is specified for L2TPv3 authentication.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

If you do not use the **hostname** command, the host name of the router is used for L2TPv3 authentication.

Examples

The following example configures the host name yb2 for a provider edge router used at one end of an L2TPv3 control channel in a L2TPv3 pseudowire configured using the L2TP class configuration named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# hostname yb2
```

Related Commands

Command	Description
ip local interface	Configures the IP address of the PE router interface to be used as the source IP address for sending tunneled packets.
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

ip dfbit set

To enable the Don't Fragment (DF) bit in the outer Layer 2 Tunnel Protocol Version 3 (L2TPv3) header, use the **ip dfbit set** command in pseudowire class configuration mode. To disable the DF bit setting, use the **no** form of this command.

ip dfbit set

no ip dfbit set

Syntax Description

This command has no arguments or keywords.

Defaults

The default value is DF bit off, except for Cisco 12000 series Internet routers, which have this command enabled by default.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use this command to set the DF bit on if, for performance reasons, you do not want tunneled packet reassembly to be performed on the router.

Note The **no ip dfbit set** command is not supported on the Cisco 12000 series Internet routers.

Examples

The following example shows how to enable the DF bit in the header of outer L2TPv3 header in pseudowires created from the pseudowire class named ether-pw:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# ip dfbit set
```

Related Commands

Command	Description
ip pmtu	Enables the discovery of a PMTU for L2TPv3 traffic.
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

ip local interface

To configure the IP address of the provider edge router interface to be used as the source IP address for sending tunneled packets, use the **ip local interface** command in pseudowire class configuration mode. To remove the IP address, use the **no** form of this command.

ip local interface *interface-name*

no ip local interface *interface-name*

Syntax Description

interface-name	Name of the PE interface whose IP address is used as the source IP address for sending tunneled packets over an Layer 2 Tunnel Protocol Version 3 (L2TPv3) pseudowire.

Defaults

No default behavior or values.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the same local interface name for all pseudowire classes configured between a pair of PE routers. It is highly recommended that a loopback interface is configured with this command. If you do not configure a loopback interface, the router will choose the "best available local address," which could be any IP address configured on a core-facing interface. This configuration could prevent a control channel from being established.

Note The interface configured with the **ip local interface** command must be a loopback interface on Cisco 12000 series Internet routers.

Examples

The following example shows how to configure the IP address of the local Ethernet interface named e0/0 as the source IP address for sending Ethernet packets through an L2TPv3 session:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# ip local interface e0/0
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

ip pmtu

To enable the discovery of a path maximum transmission unit (PMTU) for Layer 2 Tunnel Protocol Version 3 (L2TPv3) traffic, use the **ip pmtu** command in pseudowire class configuration mode. To disable PMTU discovery, use the **no** form of this command.

ip pmtu

no pmtu

Syntax Description

This command has no arguments or keywords.

Defaults

Path MTU discovery is disabled.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **ip pmtu** command enables the processing of Internet Control Message Protocol (ICMP) unreachable messages that indicate fragmentation errors in the IP backbone network carrying the tunneled traffic. The MTU of the L2TPv3 session is updated according to the MTU information contained in the ICMP unreachable message.

The **ip pmtu** command also enables MTU checking for IP packets that are sent into an L2TPv3 session with the Don't Fragment (DF) bit set. If an IP packet is larger than the MTU of the tunnel, the packet is dropped and an ICMP unreachable message is sent. If an IP packet is smaller than the MTU of the tunnel, the DF bit in the packet header is reflected from the inner IP header to the tunnel header.

Examples

The following example shows how to enable the discovery of the path MTU for pseudowires created from the pseudowire class named ether-pw:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# ip pmtu
```

Related Commands

Command	Description
ip dfbit set	Enables the DF bit in the outer L2TPv3 tunnel header.
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

ip protocol

To configure the Layer 2 Tunneling Protocol (L2Tp) or Universal Tunnel Interface (UTI) as the IP protocol used for tunneling packets in an Layer 2 Tunnel Protocol Version 3 (L2TPv3) pseudowire, use the **ip protocol** command in pseudowire class configuration mode. To remove the IP protocol configuration, use the **no** form of this command.

ip protocol {**l2tp** | **uti** | *protocol-number*}

no ip protocol {**l2tp** | **uti** | *protocol-number*}

Syntax Description

l2tp	Configures L2TP as the IP protocol used to tunnel packets in an L2TPv3 pseudowire.
uti	Configures UTI as the IP protocol used to tunnel packets in an L2TPv3 pseudowire, and allows a router running L2TPv3 to interoperate with a peer running UTI.
protocol-number	The protocol number of the desired IP protocol. The protocol number for L2TPv3 is 115. The protocol number for UTI is 120.

Defaults

The default IP protocol is L2TP.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the **ip protocol** command to ensure backward compatibility with routers running UTI. This command allows you to configure an L2TPv3 pseudowire between a router running L2TPv3 and a peer router running UTI.

Note You can use the **ip protocol** command only if you have already entered the **encapsulation l2tpv3** command.

To configure L2TP as the IP protocol used to tunnel packets in an L2TPv3 pseudowire, you may enter **115**, the IP protocol number assigned to L2TPv3, instead of **l2tp** in the **ip protocol** command.

To configure UTI as the IP protocol used to tunnel packets in an L2TPv3 pseudowire, you may enter **120**, the IP protocol number assigned to UTI, instead of **uti** in the **ip protocol** command.

Note Interoperability in an L2TPv3 control channel between a router running UTI and a router configured for L2TPv3 encapsulation is supported only if you disable signaling using the **protocol none** command.

Examples

The following example shows how to configure UTI as the IP protocol used to tunnel packets in an L2TPv3 pseudowire created from the pseudowire class named ether-pw:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# encapsulation l2tpv3
Router(config-pw)# ip protocol uti
```

Related Commands

Command	Description
encapsulation l2tpv3	Configures L2TPv3 as the data encapsulation method used to tunnel IP traffic.
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

ip tos

To configure the Type of Service (ToS) byte in the header of Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunneled packets, use the **ip tos** command in pseudowire class configuration mode. To disable a configured ToS value or IP ToS reflection, use the **no** form of this command.

ip tos {**value** *value* | **reflect**}

no tos {**value** *value* | **reflect**}

Syntax Description

value *value*	Sets the value of the ToS byte for IP packets in an L2TPv3 session. Valid values range from 0 to 255. The default value is 0.
reflect	Sets the value of the ToS byte for IP packets in an L2TPv3 session to be reflected from the inner IP header.

Defaults

The default ToS value is 0.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **ip tos** command allows you to manually configure the value of the ToS byte used in the headers of L2TPv3 tunneled packets or to have the ToS value reflected from the IP header of the encapsulated packet.

Note The **reflect** option is not supported on the Cisco 10720 and Cisco 12000 series Internet routers.

Note IP ToS byte reflection only functions if traffic in an L2TPv3 session carries IP packets as its payload.

In addition, you can configure both IP ToS reflection and a ToS priority level (from 0 to 255) for a pseudowire class. In this case, the ToS value in the tunnel header defaults to the value you specify with the **ip tos** *value* command. IP packets received on the Layer 2 interface and encapsulated into the L2TPv3 session have their ToS byte reflected into the outer IP session, overriding the default value configured with the **ip tos** *value* command.

Examples

The following example shows how to configure the ToS byte in the headers of tunneled packets in L2TPv3 tunnels created from the pseudowire class named ether-pw to be reflected from the ToS value in the header of each encapsulated IP packet:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# ip tos reflect
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

ip ttl

To configure the time-to-live (TTL) byte in the IP headers of Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunneled packets, use the **ip ttl** command in pseudowire class configuration mode. To remove the configured TTL value, use the **no** form of this command.

ip ttl *value*

no ip ttl *value*

Syntax Description

value	Value of the TTL byte in the IP headers of L2TPv3 tunneled packets. The valid values range from 1 to 255. The default value is 255.

Defaults

The default value of the TTL byte is 255.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use this command to set the Don't Fragment (DF) bit on if, for performance reasons, you do not want tunneled packet reassembly to be performed on the router.

Examples

The following example shows how to set the TTL byte to 100 in the IP header of L2TPv3 tunneled packets in pseudowires created from the pseudowire class named ether-pw:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# ip ttl 100
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

l2tp-class

To create a template of Layer 2 Tunneling Protocol control plane configuration settings that can be inherited by different pseudowire classes and to enter L2TP class configuration mode, use the **l2tp-class** command in global configuration mode.

l2tp-class [*l2tp-class-name*]

Syntax Description

l2tp-class-name	(Optional) Name of the L2TP class. The *l2tp-class-name* argument must be specified if you want to configure multiple sets of L2TP control parameters.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **l2tp-class** *l2tp-class-name* command allows you to configure an L2TP class template that consists of configuration settings used by different pseudowire classes. An L2TP class includes the following configuration settings:

- Host name of local router used during L2TPv3 authentication
- Authentication enabled
- Time interval used to exchange hello packets
- Password used for control channel authentication
- Packet size of receive window
- Retransmission settings for control packets
- Time allowed to set up a control channel

The **l2tp-class** command enters L2TP class configuration mode, where L2TP control plane parameters are configured.

You must use the same L2TP class in the pseudowire configuration at both ends of an L2TPv3 control channel.

Examples

The following example shows how to switch to L2TP class configuration mode to create an L2TP class configuration template for the class named ether-pw:

```
Router(config)# l2tp-class ether-pw
```

Related Commands

Command	Description
protocol l2tpv3	Specifies that L2TPv3 is the signaling protocol to be used to manage the pseudowires created from a pseudowire class for a dynamic L2TPv3 session, and that control plane configuration settings are to be taken from the specified L2TP class
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

l2tp cookie local

To configure the size of the cookie field used in the Layer 2 Tunnel Protocol Version 3 (L2TPv3) headers of incoming packets received from the remote PE peer router, use the **l2tp cookie local** command in xconnect configuration mode. To remove the configured cookie field parameters, use the **no** form of this command.

l2tp cookie local *size low-value* [*high-value*]

no l2tp cookie local *size low-value* [*high-value*]

Syntax Description

size	The size of the cookie field in L2TPv3 headers. The valid values are 0, 4, and 8.
low-value	The value of the lower 4 bytes of the cookie field.
high-value	(Optional) The value of the upper 4 bytes of the cookie field. For 8-byte cookie fields, you must enter the value for the upper four bytes of the cookie field.

Defaults

No cookie value is included in the header of L2TP packets.

Command Modes

Xconnect configuration mode

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **l2tp cookie local** command specifies the values that the peer PE router includes in the cookie field in L2TPv3 headers of the packets it sends to the local PE router through an L2TPv3 session. These values are required in a static L2TPv3 session.

The cookie field is an optional part of an L2TPv3 header with a length of either 4 or 8 bytes. If you specify an 8-byte length, you must also enter a value for the *high-value* argument.

Note For the Cisco 10720 and Cisco 12000 series Internet routers, an 8-byte cookie must be configured with this command.

Examples

The following example shows how to configure the cookie field of 4 bytes starting at 54321 for the L2TPv3 headers in incoming tunneled packets sent from the remote PE peer:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn)# l2tp cookie local 4 54321
```

Related Commands

Command	Description
l2tp cookie remote	Configures the size of the cookie field used in the L2TPv3 headers of outgoing (sent) packets from the remote PE peer router.
l2tp hello	Configures the interval used between sending hello keepalive messages.
l2tp id	Configures the IDs used by the local and remote PE routers at each end of an L2TPv3session.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

l2tp cookie remote

To configure the size of the cookie field used in the Layer 2 Tunnel Protocol Version 3 (L2TPv3) headers of outgoing packets sent from the local provider edge peer router, use the **l2tp cookie remote** command in xconnect configuration mode. To remove the configured cookie field parameters, use the **no** form of this command.

l2tp cookie remote *size low-value* [*high-value*]

no l2tp cookie remote *size low-value* [*high-value*]

Syntax Description

size	The size of the cookie field in L2TPv3 headers. The valid values are 0, 4, and 8.
low-value	The value of the lower 4 bytes of the cookie field.
high-value	(Optional) The value of the upper 4 bytes of the cookie field. For 8-byte cookie fields, you must enter the value for the upper four bytes of the cookie field.

Defaults

No cookie value is included in the header of L2TP packets.

Command Modes

Xconnect configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **l2tp cookie local** command specifies the values that the local PE router includes in the cookie field in L2TPv3 headers of the packets it sends to the remote PE router through an L2TPv3 session. These values are required in a static L2TPv3 session.

The cookie field is an optional part of an L2TPv3 header with a length of either 4 or 8 bytes. If you specify an 8-byte length, you must also enter a value for the *high-value* argument.

Examples

The following example shows how to configure the cookie field of 4 bytes starting at 12345 for the L2TPv3 headers in outgoing tunneled packets sent to the remote PE peer:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn)# l2tp cookie remote 4 12345
```

Related Commands

Command	Description
l2tp cookie local	Configures the size of the cookie field used in the L2TPv3 headers of incoming (received) packets from the remote PE peer router.
l2tp hello	Configures the interval used between sending hello keepalive messages.
l2tp id	Configures the IDs used by the local and remote PE routers at each end of an L2TPv3 session.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

l2tp hello

To specify the use of a hello keepalive setting contained in a specified Layer 2 Tunneling Protocol class configuration for a static Layer 2 Tunnel Protocol Version 3 (L2TPv3) session, use the **l2tp hello** command in xconnect configuration mode. To disable the sending of hello keepalive messages, use the **no** form of this command.

l2tp hello *l2tp-class-name*

no l2tp hello *l2tp-class-name*

Syntax Description

l2tp-class-name	Specifies the L2TP class configuration in which the hello keepalive interval to be used for the L2TPv3 session is stored.

Defaults

No hello keepalive messages are sent.

Command Modes

Xconnect configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Because a static L2TPv3 session does not use a control plane to dynamically negotiate control channel parameters, you must use the **l2tp hello** command to specify an L2TP class configuration that contains the interval for sending hello keepalive messages.

Examples

The following example shows how to configure the time interval for hello keepalive messages stored in the L2TP class configuration named l2tp-defaults for an Ethernet interface using the configuration settings stored in the pseudowire class named ether-pw:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn)# l2tp hello lt2p-defaults
```

Related Commands

Command	Description
l2tp cookie local	Configures the size of the cookie field used in the L2TPv3 headers of incoming (received) packets from the remote PE peer router.
l2tp cookie remote	Configures the size of the cookie field used in the L2TPv3 headers of outgoing (transmitted) packets from the remote PE peer router.
l2tp id	Configures the IDs used by the local and remote PE routers at each end of an L2TPv3 session.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

l2tp id

To configure the identifiers used by the local and remote provider edge routers at each end of a Layer 2 Tunnel Protocol Version 3 (L2TPv3) session, use the **l2tp id** command in Xconnect configuration mode. To remove the configured identifiers for local and remote sessions, use the **no** form of this command.

l2tp id *local-session-ID remote-session-ID*

no l2tp id *local-session-ID remote-session-ID*

Syntax Description

local-session-ID	The identifier used by the local PE router for as its local session identifier.
remote-session-ID	The identifier used by the remote PE router as its local session identifier.

Defaults

No default behavior or values.

Command Modes

Xconnect configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The Xconnect configuration that binds an attachment circuit to an L2TPv3 pseudowire is not complete without configured values for the *local-session-ID* and *local-session-ID* arguments.

Examples

The following example shows to configure the identifiers named 222 for the local PE router and 111 for the remote peer in an L2TPv3 session bound to an Ethernet circuit using the L2TPv3 configuration settings stored in the pseudowire class named ether-pw:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn)# l2tp id 222 111
```

Related Commands

Command	Description
l2tp cookie local	Configures the size of the cookie field used in the L2TPv3 headers of incoming (received) packets from the remote PE peer router.
l2tp cookie remote	Configures the size of the cookie field used in the L2TPv3 headers of outgoing (transmitted) packets from the remote PE peer router.
l2tp hello	Configures the interval used between sending hello keepalive messages.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

password

To configure the password used by a provider edge router for Layer 2 Tunnel Protocol Version 3 (L2TPv3) authentication, use the **password** command in L2TP class configuration mode. To disable a configured password, use the **no** form of this command.

password [*encryption-type*] *password*

no password [*encryption-type*] *password*

Syntax Description

encryption-type	(Optional) Specifies the type of encryption to use. The valid values are from 0 to 7. Currently defined encryption types are 0 (no encryption) and 7 (text is encrypted using an algorithm defined by Cisco).
password	Specifies the password used for L2TPv3 authentication.

Defaults

If a password is not configured for the L2TP class with the **password** command, the password configured with the **username password** command in global configuration mode is used.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The password that you define with the **password** command is also used for AVP hiding.

The password hierarchy sequence used for a local and remote peer PE for L2TPv3 authentication is as follows:

- The L2TPv3 password (configured with the **password** command) is used first.
- If no L2TPv3 password exists, the globally configured password (configured with the **username password** command) for the router is used.

Examples

The following example sets the password named tunnel2 to be used to authenticate an L2TPv3 session between the local and remote peers in L2TPv3 pseudowires configured with the L2TP class configuration named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# password tunnel2
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

protocol

To specify the signaling protocol to be used to manage the pseudowires created from a pseudowire class for a dynamic Layer 2 Tunnel Protocol Version 3 (L2TPv3) session, and that control plane configuration settings are to be taken from a specified L2TP class, use the **protocol** command in pseudowire class configuration mode. To remove the signaling protocol (and the control plane configuration to be used) for a pseudowire class, use the **no** form of this command.

protocol {**l2tpv3** | **none**}[*l2tp-class-name*]

no protocol {**l2tpv3** | **none**} [*l2tp-class-name*]

Syntax Description

l2tpv3	Specifies that L2TPv3 signaling protocol will be used in L2TPv3 sessions.
none	Specifies that no signaling protocol will be used in L2TPv3 sessions.
l2tp-class-name	(Optional) The name of the L2TP class whose control plane configuration is to be used for pseudowires in dynamic L2TPv3 sessions set up from a specified pseudowire class.

Defaults

The default protocol option is **l2tpv3**.

If you do not enter a value for the *l2tp-class-name* argument, the default control plane configuration settings in the L2TP signaling protocol are used.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use the **protocol l2tpv3** command to configure L2TPv3 as the signaling protocol to use in dynamic L2TPv3 sessions created from the specified pseudowire class. In addition, you can use this command to specify the L2TP class (see the section "Configuring the Xconnect Attachment Circuit") from which the control plane configuration settings are to be taken for use in a dynamic L2TPv3 session.

Use the **protocol none** command to specify that no signaling will be used in L2TPv3 sessions created from the specified pseudowire class. This configuration is required for interoperability with a remote peer running the Universal Tunnel Interface (UTI).

Do not use the command if you want to configure a pseudowire class used to create manual L2TPv3 sessions (see the section "Static L2TPv3 Sessions").

Examples

The following example shows how to switch to pseudowire configuration mode, and how to configure L2TPv3 as the signaling protocol. The control plane configuration used in the L2TP class named class1 will be used to create dynamic L2TPv3 sessions for a VLAN Xconnect interface:

```
Router(config)# pseudowire-class vlan-xconnect
Router(config-pw)# protocol l2tpv3 class1
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

pseudowire-class

To specify the name of a Layer 2 Tunnel Protocol Version 3 (L2TPv3) pseudowire class and enter pseudowire class configuration mode, use the **pseudowire-class** command in global configuration mode.

pseudowire-class [*pw-class-name*]

Syntax Description

pw-class-name	(Optional) The name of a L2TP pseudowire class. If you want to configure more than one pseudowire class, you must enter a value for the *pw-class-name* argument.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The **pseudowire-class** command allows you to configure a pseudowire class template that consists of configuration settings used by all attachment circuits bound to the class. A pseudowire class includes includes the following configuration settings:

- Data encapsulation type
- Control protocol
- Sequencing
- IP address of the local L2TPv3 interface
- Type of Service (ToS) value in IP headers

After you enter the **pseudowire-class** command, you switch to pseudowire class configuration mode, where pseudowire settings may be configured.

Examples

The following example shows how to switch to pseudowire class configuration mode to configure a pseudowire configuration template named ether-pw:

```
Router(config)# pseudowire-class ether-pw
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.
xconnect	Binds an attachment circuit to an L2TPv3 pseudowire for Xconnect service and enters xconnect configuration mode.

receive-window

To configure the packet size of the receive window on the remote provider edge router at the other end of an Layer 2 Tunnel Protocol Version 3 (L2TPv3) control channel, use the **receive-window** command in L2TP class configuration mode. To disable the configured value, use the **no** form of this command.

receive-window *size*

no receive-window *size*

Syntax Description

size	The number of packets that can be received by the remote peer before backoff queueing occurs. The valid values range from 1 to the upper limit the peer has for receiving packets. The default value is the upper limit.

Defaults

The default value is the upper limit the remote peer has for receiving packets.

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

To determine the upper limit for the *size* argument, refer to the platform-specific documentation for the peer router.

Examples

The following example sets a receive window of 30 packets to the remote peer in L2TPv3 pseudowires configured with the L2TP class named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# receive-window 30
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

retransmit

To configure the retransmission settings of control packets, use the **retransmit** command in L2TP class configuration mode. To disable the configured values, use the **no** form of this command.

retransmit {**initial retries** *initial-retries* | **retries** *retries* | **timeout** {**max** | **min**} *timeout*}

no retransmit {**initial retries** *initial-retries* | **retries** *retries* | **timeout** {**max** | **min**} *timeout*}

Syntax Description

initial retries *initial-retries*	Specifies how many start control channel requests (SCCRQs) are resent before giving up on the session. Valid values for the *initial-retries* argument range from 1 to 1000. The default value is 2
retries *retries*	Specifies how many retransmission cycles occur before determining that the peer PE router does not respond. Valid values for the *retries* argument range from 1 to 1000. The default value is 15.
timeout {**max**	**min**} *timeout*	Specifies maximum and minimum retransmission intervals (in seconds) for retransmitting control packets. Valid values for the *timeout* argument range from 1 to 8. The default maximum interval is 8; the default minimum interval is 1.

Defaults

Initial retries: 2
Retries: 15
Maximum timeout interval: 8 seconds
Minimum timeout interval: 1 second

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use this command to configure the amount of time spent trying to establish or maintain a control channel.

Examples

The following example configures ten retries for sending Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunneled packets to a remote peer in L2TPv3 pseudowires configured with the L2TP class named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# retransmit retries 10
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

sequencing

To configure the direction in which sequencing is enabled for data packets in an L2TPv3 pseudowire, use the **sequencing** command in pseudowire class configuration mode. To remove the sequencing configuration from the pseudowire class, use the **no** form of this command.

sequencing {**transmit** | **receive** | **both**}

no sequencing {**transmit** | **receive** | **both**}

Syntax Description

transmit	Updates the Sequence Number field in the headers of data packets sent over the pseudowire according to the data encapsulation method that is used.
receive	Keeps the value in the Sequence Number field in the headers of data packets received over the pseudowire. Out-of-order packets are dropped.
both	Enables both the **transmit** and **receive** options.

Defaults

Sequencing is off.

Command Modes

Pseudowire class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

When you enable sequencing using any of the available options, L2TPv3 automatically enables the sending of sequence numbers and requests the remote PE peer to send sequence numbers. Out-of-order packets received on the pseudowire are dropped only if you use the **sequencing receive** or **sequencing both** commands.

In Cisco IOS Release 12.0(23)S, sequencing is not supported on the Cisco 10720 Internet router and the Cisco 12000 series Internet routers. If the L2TPv3 peer router requests sequence numbers for an L2TPv3 session configured on a Cisco 10720 Internet router or Cisco 12000 series Internet router, the request to establish the session is denied.

If sequencing is enabled for L2TPv3 pseudowires on the Cisco 7500 series, all traffic on the pseudowires is switched through the Route Switch Processor (RSP) regardless of the setting configured with the **ip cef distributed** command.

Examples

The following example shows how to enable sequencing in data packets in L2TPv3 pseudowires created from the pseudowire class named ether-pw so that Sequence Number field is updated in tunneled packet headers for data packets both sent and received over the pseudowire:

```
Router(config)# pseudowire-class ether-pw
Router(config-pw)# sequencing both
```

Related Commands

Command	Description
pseudowire-class	Specifies the name of an L2TP pseudowire class and enters pseudowire class configuration mode.

show l2tun session

To display the current state of a Layer 2 session and display protocol information about a Layer 2 Tunnel Protocol Version 3 (L2TPv3) control channel, use the **show l2tun session** command in EXEC configuration mode.

show l2tun session [**all** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **brief** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **circuit** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **l2tp** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **packets** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **sequence** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*] | **state** [**ip-addr** *ip-address* [**vcid** *number*] | **vcid** *number*]]

Syntax Description

all	(Optional) Displays information about all current L2TPv3 sessions on the router.
ip-addr *ip-address*	(Optional) IP address of interface of the peer provider edge (PE) router on which one or more L2TPv3 sessions have been configured. Specifies the IP address of the peer PE router and the 32-bit virtual circuit identifier shared between the PE at each end of the control channel. The peer router ID (IP address) and virtual circuit identifier must be a unique combination on the router.
vcid *number*	(Optional) 32-bit virtual circuit identifier shared between the peer PE and the local router at each end of the control channel.
brief	(Optional) Displays information about all current L2TPv3 sessions, including peer ID address and circuit status of the L2TPv3 sessions.
circuit	(Optional) Displays information about all current L2TPv3 sessions, including circuit status (up or down).
l2tp	(Optional) Displays information about L2TP for all current L2TPv3 sessions.
packets	(Optional) Displays information about the packet counters (in and out) associated with current L2TPv3 sessions.
sequence	(Optional) Displays sequencing information about each L2TPv3 session, including number of out-of-order and returned packets.
state	(Optional) Displays information about all current L2TPv3 sessions and their protocol state, including remote virtual circuit identifier s.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

When you use the **show l2tun session** command to display information about current L2TPv3 sessions on the router, you can filter the output as follows:

- To filter the output to include only L2TPv3 sessions set up for a specific IP address, enter **ip-addr** *ip-address* in the command.
- To filter the output to include only the L2TPv3 session that matches the specified remote IP address and virtual circuit identifier, enter **ip-addr** *ip-address* **vcid** *number* in the command.
- To filter the output to include only L2TPv3 sessions set up for a specific IP address, enter **vcid** *number* in the command.

Examples

The following example shows how to display detailed information about all current L2TPv3 sessions:

```
Router# show l2tun session all
 Session Information Total tunnels 1 sessions 1

Session id 32519 is up, tunnel id 30866
Call serial number is 2074900019
Remote tunnel name is tun1
  Internet address is 172.18.184.142
  Session is L2TP signalled
  Session state is established, time since change 03:02:36
    10001 Packets sent, 9936 received
    1180064 Bytes sent, 1172390 received
  Session vcid is 200
  Session Layer 2 circuit, type is Ethernet Vlan, name is FastEthernet0/1/0.2:2
  Circuit state is UP
    Remote session id is 18818, remote tunnel id 6809
  Set DF bit to 0
  Session cookie information:
    local cookie, size 4 bytes, value 84 B0 0A 9F
    remote cookie, size 4 bytes, value 43 46 80 A3
  SSS switching enabled
  Sequencing is on
    Ns 10001, Nr 9953, 0 out of order packets discarded
```

The following example shows how to display information only about the L2TPv3 session set up on a peer PE router with an IP address of 172.18.184.142 and a virtual circuit identifier of 300:

```
Router# show l2tun session all ip-addr 172.18.184.142 vcid 300

L2TP Session

Session id 32518 is up, tunnel id 35217
Call serial number is 2074900020
Remote tunnel name is tun1
  Internet address is 172.18.184.142

  Session is L2TP signalled
  Session state is established, time since change 03:06:39
    9932 Packets sent, 9932 received
    1171954 Bytes sent, 1171918 received
  Session vcid is 300
  Session Layer 2 circuit, type is Ethernet Vlan, name is FastEthernet0/1/0.3:3
  Circuit state is UP
    Remote session id is 18819, remote tunnel id 37340
  Set DF bit to 0
  Session cookie information:
    local cookie, size 4 bytes, value CF DC 5B F3
    remote cookie, size 4 bytes, value FE 33 56 C4
  SSS switching enabled
```

```
  Sequencing is on
    Ns 9932, Nr 10001, 0 out of order packets discarded
```

The following example shows how to display information about the circuit status of L2TPv3 sessions on a router:

```
Router# show l2tun session circuit

Session Information Total tunnels 3 sessions 3

LocID      TunID      Peer-address    Type Stat Username, Intf/
                                              Vcid, Circuit
32517      26515      172.18.184.142  VLAN UP   100, Fa0/1/0.1:1
32519      30866      172.18.184.142  VLAN UP   200, Fa0/1/0.2:2
32518      35217      172.18.184.142  VLAN UP   300, Fa0/1/0.3:3
```

Related Commands

Command	Description
show l2tun tunnel	Displays the current state of an L2TPv3 session and display information about currently configured sessions, including local and remote L2TP host names, aggregate packet counts, and L2TP control channels.

show l2tun tunnel

To display the current state of a Layer 2 Tunnel Protocol Version 3 (L2TPv3) session and display information about currently configured sessions, including local and remote L2TP host names, aggregate packet counts, and L2TP control channels, use the **show l2tun tunnel** command in EXEC mode.

show l2tun tunnel [**all** [**id** *identifier* | **local-name** *local-name remote-name* | **remote-name** *remote-name local-name*] | **packets** [**id** *identifier* | **local-name** *local-name remote-name* | **remote-name** *remote-name local-name*] | **state** [**id** *identifier* | **local-name** *local-name remote-name* | **remote-name** *remote-name local-name*] | **summary** [**id** *identifier* | **local-name** *local-name remote-name* | **remote-name** *remote-name local-name*] |**transport** [**id** *identifier* | **local-name** *local-name remote-name* | **remote-name** *remote-name local-name*]]

Syntax Description

all	(Optional) Displays information about all current L2TP sessions configured on the router.
id *identifier*	(Optional) Specifies the local tunnel ID number.
local-name *local-name remote-name*	(Optional) Specifies the local and remote names used in the L2TPv3 session.
remote-name *remote-name local-name*	(Optional) Specifies the remote and local names used in the L2TPv3 session.
packets	(Optional) Displays aggregate packet counts for all negotiated L2TPv3 sessions.
state	(Optional) Displays information about the current state of L2TPv3 sessions, including the local and remote host names for each control channel.
summary	(Optional) Displays a summary of L2TP sessions on the router and their current state, including the number of virtual private dialup network (VPDN) sessions associated with each control channel.
transport	(Optional) Displays information about the L2TP control channels used in each session and the local and remote IP addresses at each end of the control channel.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

When you use the **show l2tun tunnel** command to display information about configured L2TP sessions on the router, you can filter the output as follows:

- To filter the output to include only L2TP sessions set up using the local tunnel ID, enter **id** *identifier* in the command.
- To filter the output to include only the L2TP session that matches the specified local IP name and remote name, enter either **local-name** *local-name remote-name* or **remote-name** *remote-name local-name* in the command.

Examples

The following example shows how to display detailed information about all currently configured L2TP sessions:

```
Router# show l2tun tunnel all
 Session Information Total tunnels 1 sessions 1

Tunnel Information Total tunnels 1 sessions 1

Tunnel id 26515 is up, remote id is 41814, 1 active sessions
  Tunnel state is established, time since change 03:11:50
  Tunnel transport is IP (115)
  Remote tunnel name is tun1
    Internet Address 172.18.184.142, port 0
  Local tunnel name is Router
    Internet Address 172.18.184.116, port 0
  Tunnel domain is
  VPDN group for tunnel is
  0 packets sent, 0 received
  0 bytes sent, 0 received
  Control Ns 11507, Nr 11506
  Local RWS 2048 (default), Remote RWS 800
  Tunnel PMTU checking disabled
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 1, max 1
  Total resends 0, ZLB ACKs sent 11505
  Current nosession queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0 0
  Sessions disconnected due to lack of resources 0
```

The following example shows how to filter information to display L2TP control channel details only for the sessions configured with the local name Router and the remote name tun1:

```
Router# show l2tun tunnel transport local-name Router tun1

Tunnel Information Total tunnels 3 sessions 3

LocID Type Prot  Local Address   Port  Remote Address  Port
26515 IP   115   172.18.184.116  0     172.18.184.142  0
30866 IP   115   172.18.184.116  0     172.18.184.142  0
35217 IP   115   172.18.184.116  0     172.18.184.142  0
```

The following example shows how to display information about the current state of L2TP sessions with the local and remote host names of each session:

```
Router# show l2tun tunnel state

LocID  RemID   Local Name Remote Name  State  Last-Chg
26515  41814   Router     tun1        est     03:13:15
30866  6809    Router     tun1        est     03:13:15
35217  37340   Router     tun1        est     03:13:15
```

Related Commands

Command	Description
show l2tun session	Displays the current state of a Layer 2 session and displays protocol information about an L2TPv3 control channel.

snmp-server enable traps l2tun session

To enable Simple Network Management Protocol (SNMP) notifications (traps or inform requests) for Layer 2 Tunnel Protocol Version 3 (L2TPv3) sessions, use the **snmp-server enable traps l2tun session** command in global configuration mode. To disable SNMP notifications, use the **no** form of this command.

snmp-server enable traps l2tun session

no snmp-server enable traps l2tun session

Syntax Description

This command has no arguments or keywords.

Defaults

This command is disabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. This command enables both traps and inform requests for L2TP sessions. To specify whether the notifications should be sent as traps or informs, use the **snmp-server host** [**traps** | **informs**] command.

If you do not enter the **snmp-server enable traps l2tun session** command, no notifications are sent.

The **snmp-server enable traps l2tun session** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP notifications.

Examples

The following example enables the router to send L2TP session traps to the host specified by the name myhost.cisco.com, using the community string defined as public:

```
Router(config)# snmp-server enable traps l2tun session
Router(config)# snmp-server host myhost.cisco.com public
```

Related Commands

Command	Description
snmp-server host	Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.

timeout setup

To configure the amount of time allowed to set up a control channel with a remote PE router at the other end of a Layer 2 Tunnel Protocol Version 3 (L2TPv3) pseudowire, use the **timeout setup** command in L2TP class configuration mode. To disable the configured value, use the **no** form of this command.

timeout setup *seconds*

no timeout setup *seconds*

Syntax Description

seconds	The number seconds allowed to set up an L2TPv3 control channel. The valid values range from 60 to 6000. The default value is 300 seconds.

Defaults

300 seconds

Command Modes

L2TP class configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

Use this command to configure the amount of time spent attempting to establish a control channel.

Examples

The following example sets a timeout period of 200 seconds to establish a control channel with a remote peer in L2TPv3 pseudowires configured with the L2TP class named l2tp class1:

```
Router(config)# l2tp-class l2tp-class1
Router(config-l2tp-class)# timeout setup 200
```

Related Commands

Command	Description
l2tp-class	Creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes and enters L2TP class configuration mode.

xconnect

To bind an Ethernet, 802.1q VLAN, or Frame Relay attachment circuit to a Layer 2 Tunnel Protocol Version 3 (L2TPv3) pseudowire for Xconnect service and enter xconnect configuration mode, use the **xconnect** command in interface configuration mode.

xconnect *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]

Syntax Description

peer-ip-address	The IP address of the remote PE peer.
vcid	The 32-bit identifier of the virtual circuit between the routers at each end of the L2TPv3 control channel.
pseudowire-parameters	The encapsulation and pseudowire class parameters to be used for the L2TPv3 control channel. At least one of the following pseudowire parameters must be configured: • **encapsulation** {**l2tpv3** [**manual**] | **mpls**}—The encapsulation pseudowire class parameter specifies the tunneling method used to encapsulate data in the pseudowire: – **l2tpv3**—L2TPv3 is the tunneling method to be used. – **manual**—(Optional) No signaling is to be used in the L2TPv3 control channel. This command places the router in xconnect configuration mode for manual configuration of L2TPv3 parameters for the attachment circuit. – **mpls**—Multiprotocol Label Switching (MPLS) is the tunneling method to be used. • **pw-class** {*pw-class-name*}—The pseudowire class configuration from which the data encapsulation type (L2TPv3) will be taken.
sequencing {**transmit** | **receive** | **both**}	(Optional) The sequencing method to be used for packets received or sent in L2TP sessions: • **transmit**—Sequencing of L2TP data packets received from the L2TPv3 session. • **receive**—Sequencing of L2TP data packets sent into the L2TPv3 session. • **both**—Sequencing of L2TP data packets that are both sent and received from the L2TPv3 session.

Defaults

The default behavior is to use L2TPv3 as the data encapsulation method with sequencing off.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.

Usage Guidelines

The combination of the *peer-ip-address* and *vcid* must be unique on the router. Each Xconnect configuration must have a unique combination of *peer-ip-address* and *vcid* configuration.

Note If the remote router is a Cisco 12000 series Internet router, the *peer-ip-address* argument must specify a loopback address on that router.

The same *vcid* value that identifies the attachment circuit must be configured using the **xconnect** command on the local and remote PE router at each end of an L2TPv3 session. The virtual circuit identifier creates the binding between a pseudowire and an attachment circuit.

To manually configure the L2TP settings used in the attachment circuit, enter **encapsulation l2tpv3 manual** in the **xconnect** command. This configuration is called a static L2TPv3 session. You are switched to xconnect configuration mode and can then configure the following options:

- Local and remote session identifiers (using the **l2tp id** command) for local and remote PE routers at each end of the session.
- Size of the cookie field used in the L2TPv3 headers of incoming (sent) packets from the remote PE peer router (using the **l2tp cookie local** command).
- Size of the cookie field used in the L2TPv3 headers of outgoing (received) L2TP data packets (using the **l2tp cookie remote** command).
- Interval used between sending hello keepalive messages (using the **l2tp hello** command).

For more information about configuring a static L2TPv3 sessions, see the section "Manually Configuring L2TPv3 Session Parameters."

If you do not enter **encapsulation l2tpv3 manual** in the **xconnect** command, the data encapsulation type for the L2TPv3 session is taken from the encapsulation type configured for the pseudowire class specified with the **pw-class** *pw-class-name* command (see the section "Configuring the L2TPv3 Pseudowire").

The **pw-class** *pw-class-name* value binds the Xconnect configuration of an attachment circuit to a specific pseudowire class. In this way, the pseudowire class configuration serves as a template that contains settings used by all attachment circuits bound to it with the **xconnect** command.

Use a **pw-class** *pw-class-name* value if you need to configure more advanced options as described in the section "Configuring the Xconnect Attachment Circuit."

Note Although the **encapsulation l2tpv3** and **pw-class** *pw-classname* commands are optional, you must specify one or the other when using the **xconnect** command to bind an attachment circuit to an L2TPv3 pseudowire.

Examples

The following example shows how to configure Xconnect service for an Ethernet interface by binding the Ethernet circuit to the L2TPv3 pseudowire named 123 with a remote peer 10.0.3.201, and by using the L2TP configuration settings in the pseudowire class named vlan-xconnect:

```
Router(config)# interface Ethernet0/0.1
Router(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect
```

Related Commands

Command	Description
l2tp-class	Configures a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes.
pseudowire-class	Configures a template of pseudowire configuration settings used by the attachment circuits transported over a pseudowire.

Glossary

AVPs—attribute-value pairs.

BECN—backward explicit congestion notification. Bit set by a Frame Relay network in frames traveling in the opposite direction of frames encountering a congested path. DTE receiving frames with the BECN bit set can request that higher-level protocols take flow control action as appropriate.

CE—customer edge (Frame Relay switch or user device).

CIR—committed information rate. Rate at which a Frame Relay network agrees to transfer information under normal conditions, averaged over a minimum increment of time. CIR, measured in bits per second, is one of the key negotiated tariff metrics.

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.

dCEF—distributed Cisco Express Forwarding.

DLCI—data-link connection identifier. A unique number assigned to a PVC endpoint in a Frame Relay network. Identifies a particular PVC endpoint within an access channel in a Frame Relay network and has local significance only to that channel.

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

FECN—forward explicit congestion notification. Bit set by a Frame Relay network to inform DTE receiving the frame that congestion was experienced in the path from source to destination. DTE receiving frames with the FECN bit set can request that higher-level protocols take flow-control action as appropriate.

HDLC—High-Level Data Link Control. A generic link-level communications protocol developed by the International Organization for Standardization (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. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, whereby ISs (routers) exchange routing information based on a single metric to determine network topology.

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

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

LMI—Local Management Interface.

MQC—modular quality of service command-line interface.

MTU—maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

NNI—Network-to-Network Interface. ATM Forum standard that defines the interface between two ATM switches that are both located in a private network or are both located in a public network. The UNI standard defines the interface between a public switch and a private one. Also, the standard interface between two Frame Relay switches meeting the same criteria.

PE—Provider edge router providing Frame Relay over L2TPv3 functionality.

PPP—Point-to-Point Protocol. A link-layer encapsulation method for dialup or dedicated circuits. A successor to Serial Line IP (SLIP), PPP provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

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 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.

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

UNI—User-Network Interface.

UTI—Universal Transport 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 routers. A VPDN enables users to configure secure networks that take advantage of ISPs that tunnel remote access traffic through the ISP cloud.

WAN—Wide-area network. Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers. Frame Relay, SMDS, and X.25 are examples of WANs.