Implementing Layer 2 Tunnel Protocol Version 3

Layer 2 Tunnel Protocol Version 3 (L2TPv3) is an Internet Engineering Task Force (IETF) working group draft that provides several enhancements to L2TP, including the ability to tunnel any Layer 2 (L2) payload over L2TP. Specifically, L2TPv3 defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network using L2 virtual private networks (VPNs).

For additional information about L2TPv3, see *MPLS VPNs over IP Tunnels on Cisco IOS XR Software.*

Feature History for Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR

Release	Modification
Release 3.9.0	This feature was introduced.

Contents

- Prerequisites for Layer 2 Tunnel Protocol Version 3, page 7-165
- Information About Layer 2 Tunnel Protocol Version 3, page 7-166
- How to Implement Layer 2 Tunnel Protocol Version 3, page 7-172
- Configuration Examples for Layer 2 Tunnel Protocol Version 3, page 7-190
- Additional References, page 7-192

Prerequisites for Layer 2 Tunnel Protocol Version 3

The following prerequisites are required to implement L2TPv3:

- To perform these configuration tasks, your Cisco IOS XR software system administrator must assign you to a user group associated with a task group that includes the corresponding command task IDs. All command task IDs are listed in individual command references and in the *Cisco IOS XR Task ID Reference Guide*.

  If you need assistance with your task group assignment, contact your system administrator.
- You must enable Cisco Express Forwarding (CEF) before you configure a cross-connect attachment circuit (AC) for a customer edge (CE) device.
- 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 provider edge (PE) device at the other end of an L2TPv3 control-channel.

- You must enable Simple Network Management Protocol (SNMP) notifications of L2TP session up and session down events.

Note A cross-connection is expressed as *xconnect* in the CLI.

Information About Layer 2 Tunnel Protocol Version 3

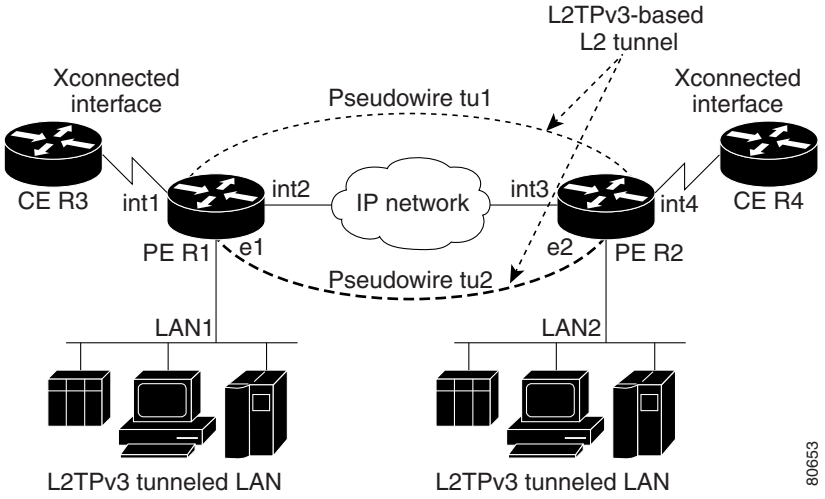
To configure the L2TPv3 feature, you should understand the following concepts:

- L2TPv3 Operation, page 7-166
- L2TPv3 Benefits, page 7-167
- L2TPv3 Features, page 7-167

L2TPv3 Operation

Figure 7-1 shows how the L2TPv3 feature is used to set 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 needn't know anything about the customer networks.

Figure 7-1 L2TPv3 Operation



In Figure 7-1, 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*. The CE routers R3 and R4 communicate through a pair of cross-connected 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 through the pseudowire control-channel (tu1*)* to R2. R2 decapsulates the packet and sends it on interface int4 to R4. When R4 needs to send a packet to R3, the packet follows the same path in reverse.

L2TPv3 Benefits

L2TPv3 provides the following benefits:

- Simplifies deployment of VPNs—L2TPv3 is an industry-standard L2 tunneling protocol that ensures interoperability among vendors, increasing customer flexibility and service availability.
- Does not require MPLS—Service providers need not deploy MPLS in the core IP backbone to set up VPNs using L2TPv3 over the IP backbone; this will result in operational savings and increased revenue.
- Supports L2 tunneling over IP for any payload—L2TPv3 provides enhancements to L2TP to support L2 tunneling of any payload over an IP core network. L2TPv3 defines the base L2TP protocol as being separate from the L2 payload that is tunneled.

L2TPv3 Features

L2TPv3 provides cross-connect support for Ethernet and 802.1q (VLAN) using the sessions described in the following sections:

- Static L2TPv3 Sessions, page 7-168
- Dynamic L2TPv3 Sessions, page 7-168

L2TPv3 also supports:

- Local Switching, page 7-168
- Local Switching: Quality of Service, page 7-169
- L2TPv3 Pseudowire Switching, page 7-169
- L2TPv3 Pseudowire Manager, page 7-169
- IP Packet Fragmentation, page 7-170
- L2TPv3 Type of Service Marking, page 7-170
- Keepalive, page 7-170
- Maximum Transmission Unit Handling, page 7-171
- Distributed switching
- L2TPv3 L2 fragmentation
- L2TPv3 control message hashing
- L2TPv3 control message rate limiting
- L2TPv3 digest secret graceful switchover
- Manual clearing of L2TPv3 tunnels
- L2TPv3 tunnel management
- Color aware policer on ethernet over L2TPv3
- Site of origin for BGP VPNs
- IPSec Mapping to L2TPv3, page 7-171
- Like-to-Like Pseudowires, page 7-171

Static L2TPv3 Sessions

Typically, the L2TP control plane is responsible for negotiating session parameters (such as the session ID or the cookie) to set up the session; however, some IP networks require sessions to be configured so that no signaling is required for session establishment. Therefore, you can 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 L2 traffic as soon as the AC 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.

Dynamic L2TPv3 Sessions

A dynamic L2TP session is established through the exchange of control messages containing attribute-value pair (AVP). Each AVP contains information about the nature of the L2 link being forwarded: including the payload type, virtual circuit (VC) ID, and so on.

Multiple L2TP sessions 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.

Local Switching

An AC to AC cross-connect, also called *local switching*, is a building block of L2VPN that allows frames to switch between two different ACs on the same PE. PE (see Figure 7-2).

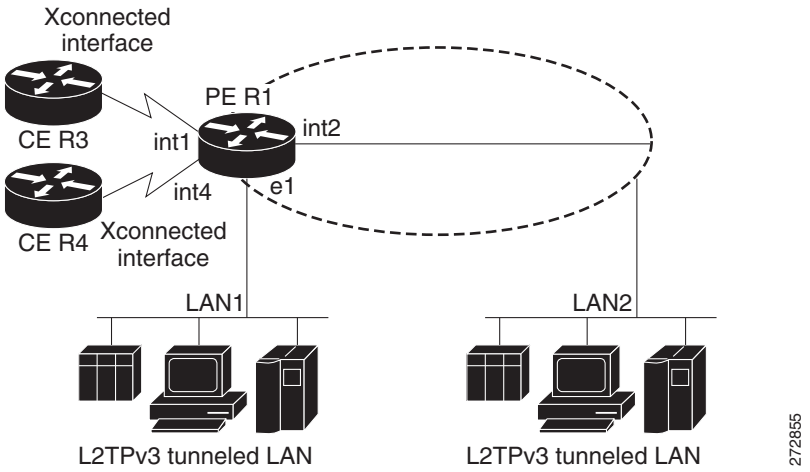
You must configure separate IP addresses for each cross-connect statement on the Carrier Edge router.

The following configurations are supported for local switching:

- Port-to-Port
- Dot1q-to-Dot1q
- QinQ-to-QinQ
- QinAny-to-QinAny
- Dot1q-to-QinQ
- QinQ-to-Dot1q
- QinQ-to-QinAny
- QinAny-to-QinQ

Note VLAN-to-VLAN options do not require interworking. If both interfaces are Ethernet VLAN, each reside on a single physical interface. By definition, local switching is not a pseudowire technology, because signaling protocols (such as LDP or L2TPv3) are not involved.

Figure 7-2 Local Switching Operation



Local Switching: Quality of Service

The following quality of service (QoS) requirements apply to local switching:

- QoS service policies can be attached directly to the AC.
- QoS service policies can be attached to the main interface using **match vlan** on L2 VLAN ACs.
- QoS service policies attached to the main interface can be inherited by all L2 VLANs.
- QoS service policies cannot be attached to a main interface when there are service policies already attached to its L3VLANs or L2VLAN ACs.
- QoS service policies already attached to the main interface are not permitted on L3 VLAN or L2 VLAN ACs.

L2TPv3 Pseudowire Switching

L2VPN pseudowire switching allows you to:

- Extend L2VPN pseudowires across an Inter-AS boundary.
- Connect two or more contiguous pseudowire segments to form an end-to-end multihop pseudowire.
- Keep the IP addresses of the edge PE routers private across Inter-AS boundaries.
- Keep different administrative or provisioning domains to manage the end-to-end service.

L2TPv3 Pseudowire Manager

The pseudowire manager is a client library provided by the pseudowire signaling module that runs in the context of the L2VPN process. This client library implements interface to pseudo-wire signaling protocol for specific pseudowire type.

IP Packet Fragmentation

It is desirable to avoid fragmentation issues in the service provider network because reassembly is computationally expensive. The easiest way to avoid fragmentation issues is to configure the CE routers with an Maximum Transmission Unit (MTU) value that is smaller than the pseudowire path MTU. However, in scenarios where this is not an option, fragmentation issues must be considered. Previously, L2TP supported only the following options for packet fragmentation when a packet is determined to exceed the L2TP path MTU:

- Unconditionally drop the packet
- Fragment the packet after L2TP/IP encapsulation
- Drop the packet and send an Internet Control Message Protocol (ICMP) unreachable message back to the CE router

Currently, the following options for packet fragmentation are supported:

- Path MTU is a configurable value which is configured on PE. If the packet size and the L2TP header size are larger than the configured path MTU, packets are dropped.
- The PE configuration requires that a backbone facing interface's MTU is always greater or equal to the customer facing interface’s MTU and L2TP header size.
- IP fragmentation is not supported with L2TPv3.

L2TPv3 Type of Service Marking

When L2 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 L2 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.

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.

Maximum Transmission Unit Handling

It is important that you configure an maximum transmission unit (MTU) appropriate for a each L2TPv3 tunneled link. The configured MTU size ensures that the lengths of the tunneled L2 frames fall below the MTU of the destination AC.

L2TPv3 handles the MTU as follows:

- Configure the path MTU on the PE. If the packet size and the L2TP header collectively are larger than the configured value, packets are dropped.

IP Security Mapping to L2 Tunneling Protocol, Version 3

Note This feature is supported only on the Cisco IPSec VPN SPA.

The L2TPv3 is a protocol that is used to tunnel a variety of payload types over IP networks. IP security (IPSec) provides an additional level of protection at a service PE router than relying on access control list (ACL) filters. L2TPv3 tunnels are also secured by using IPSec, as specified in RFC3931.

You can secure L2TPv3 tunnels by using IPSec, which provides authentication, privacy protection, integrity checking, and replay protection. When using IPSec, the tunnel head and the tunnel tail can be treated as the endpoints of an SA. A single IP address of the tunnel head is used as the source IP address, and a single IP address of the tunnel tail is used as the destination IP address.

The following scenarios are described to have L2TPv3 work with IPSec:

- IPSec Mapping to L2TPv3, page 7-171
- IPSec over L2TPv3, page 7-171

IPSec Mapping to L2TPv3

A CE 1 router sends an IPSec packet to a PE1 router. The PE1 router sends an IPSec packet to the Cisco IPSec VPN SPA by routing the look up for the front door virtual routing and forwarding (FVRF) in the service-ipsec interface. The Cisco IPSec VPN SPA can decapsulate an IPSec packet to obtain a clear IP packet, and perform a routing look up for the inside virtual routing and forwarding (IVRF) in the service-ipsec interface.

IPSec over L2TPv3

If the packet arrives at PE1 outside of a virtual routing and forwarding (VRF), for example, the global table, the packet is forwarded to the PE2 according to the global FIB in PE1. This is normal for IP switching until the packet arrives at PE2 with no encapsulation at any point.

Like-to-Like Pseudowires

A PseudoWire (PW) is a bidirectional virtual circuit (VC) connecting two Attached Circuits (ACs). In an MPLS network, PWs are carried inside an LSP tunnel.

The following features describe the pseudowire connection:

- PPP, page 7-172

PPP

A point-to-point (PPP) connection allows service providers to provide a transparent PPP pass-through where the customer-edge routers can exchange the traffic through an end-to-end PPP session. Service providers can offer a virtual leased-line solution, and use the PPP subinterface capability to peer with multiple providers through a single connection.

Note In an MPLS network, pseudowires are carried inside an LSP tunnel.

How to Implement Layer 2 Tunnel Protocol Version 3

This section includes the tasks required to implement L2TPv3, as follows:

- Configuring a Pseudowire Class, page 7-172 (required)
- Configuring L2TP Control-Channel Parameters, page 7-174 (required)
- Configuring L2TPv3 Pseudowires, page 7-184 (required)

Configuring a Pseudowire Class

Perform this task to configure a pseudowire class, or template.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **pw-class** *class name*
4. **encapsulation** {**mpls** | **l2tpv3**}
5. **protocol l2tpv3 class** *class name*
6. **ipv4 source** *ip-address*
7. **transport mode** {**ethernet** | **vlan**}
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enter L2VPN configure submode.
Step 3	**pw-class** *class name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# pw-class wkg`	Enters a pseudowire-class name.
Step 4	**encapsulation l2tpv3** **Example:** `RP/0/RP0/CPU0:router(config-l2tp-pwc)# encapsulation l2tpv3`	Configures pseudowire encapsulation to the Layer 2 Tunnel Protocol.
Step 5	**protocol l2tpv3 class** *class name* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# protocol l2tpv3 class Class_l2tp_01`	Configures the L2TPv3 dynamic pseudowire signaling protocol to be used to manage the pseudowires created. **Note** Ensure that the L2TPv3 class name begins with a letter (A to Z or a to z). The class name can contain letters (A to Z or a to z) or numbers (0 to 9) and other characters such as underscore (_), hyphen (-) or period (.). A maximum of 31 characters can be used in the class name.
Step 6	**ipv4 source** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# ipv4 source 126.10.1.55`	Configures the local source IPv4 address.

	Command or Action	Purpose
Step 7	**transport-mode** {**ethernet** | **vlan**} **Example:** `RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# transport-mode ethernet`	Configures the remote transport mode.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring L2TP Control-Channel Parameters

This section describes the tasks you must perform to create a template of L2TP control-channel parameters that can be inherited by different pseudowire classes. The three main parameters described are:

- Timing parameters
- Authentication parameters
- Maintenance parameters

L2TP control-channel parameters are used in control-channel authentication, keepalive messages, and control-channel negotiation. In a L2tpv3 session, the same L2tp class must be configured on both PE routers.

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

- Configuring L2TP Control-Channel Timing Parameters, page 7-175
- Configuring L2TPv3 Control-Channel Authentication Parameters, page 7-176
- Configuring L2TP Control-Channel Maintenance Parameters, page 7-183

Note When 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.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2tp-class** *l2tp-class-name* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class cisco`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 3	**receive-window** *size* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# receive-window 30`	Configures the number of packets that can be received by the remote peer before backoff queueing occurs. The default value is 512.

	Command or Action	Purpose
Step 4	**retransmit** {**initial retries** *initial-retries* | **retries** *retries* | **timeout** {**max** | **min**} *timeout*} **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# retransmit retries 10`	Configures parameters that affect the retransmission of control packets. • **initial retries**—Specifies how many SCCRQs are re-sent before giving up on the session. Range is 1 to 1000. The default is 2. • **retries**—Specifies how many retransmission cycles occur before determining that the peer PE router does not respond. Range is 1 to 1000. The default is 15. • **timeout** {**max** | **min**}—Specifies maximum and minimum retransmission intervals (in seconds) for resending control packets. Range is 1 to 8. The default maximum interval is 8; the default minimum interval is 1.
Step 5	**timeout setup** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# timeout setup 400`	Configures the amount of time, in seconds, allowed to set up a control-channel. • Range is 60 to 6000. Default value is 300.

Configuring L2TPv3 Control-Channel Authentication Parameters

Two methods of control-channel message authentication are available:

- L2TP Control-Channel (see Configuring Authentication for the L2TP Control-Channel, page 7-177)
- L2TPv3 Control Message Hashing (see Configuring L2TPv3 Control Message Hashing, page 7-178)

You can enable both methods of authentication to ensure interoperability with peers that support only one of these methods of authentication, but this configuration will yield control of which authentication method is used to the peer PE router. Enabling both methods of authentication should be considered an interim solution to solve backward-compatibility issues during software upgrades.

The principal difference between the L2TPv3 Control Message Hashing feature and CHAP-style L2TP control-channel authentication is that, instead of computing the hash over selected contents of a received control message, the L2TPv3 Control Message Hashing feature uses the entire message in the hash. In addition, instead of including the hash digest in only the SCCRP and SCCCN messages, it includes it in all messages.

This section also describes how to configure L2TPv3 digest secret graceful switchover (see Configuring L2TPv3 Digest Secret Graceful Switchover, page 7-179,) which lets you make the transition from an old L2TPv3 control-channel authentication password to a new L2TPv3 control-channel authentication password without disrupting established L2TPv3 tunnels.

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

Configuring Authentication for the L2TP Control-Channel

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

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

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

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

SUMMARY STEPS

1. **configure**
2. **l2tp-class** *word*
3. **authentication**
4. **password** {**0** | **7**} *password*
5. **hostname** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2tp-class** *word* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 3	**authentication** **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# authentication`	Enables authentication for the control-channel between PE routers.

	Command or Action	Purpose
Step 4	**password** {**0** | **7**} *password* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# password 7 cisco`	Configures the password used for control-channel authentication. • [**0** | **7**]—Specifies the input format of the shared secret. The default value is **0**. – **0**—Specifies an encrypted password will follow. – **7**—Specifies an unencrypted password will follow. • *password*—Defines the shared password between peer routers.
Step 5	**hostname** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# hostname yb2`	Specifies a hostname used to identify the router during L2TP control-channel authentication. • If you do not use this command, the default hostname of the router is used.

Configuring L2TPv3 Control Message Hashing

Perform this task to configure L2TPv3 Control Message Hashing feature for an L2TP class.

L2TPv3 control message hashing incorporates authentication or integrity check for all control messages. This per-message authentication is designed to guard against control message spoofing and replay attacks that would otherwise be trivial to mount against the network.

Enabling the L2TPv3Control Message Hashing feature will impact performance during control-channel and session establishment because additional digest calculation of the full message content is required for each sent and received control message. This is an expected trade-off for the additional security afforded by this feature. In addition, network congestion may occur if the receive window size is too small. If the L2TPv3 Control Message Hashing feature is enabled, message digest validation must be enabled. Message digest validation deactivates the data path received sequence number update and restricts the minimum local receive window size to 35.

You can configure control-channel authentication or control message integrity checking; however, control-channel authentication requires participation by both peers, and a shared secret must be configured on both routers. Control message integrity check is unidirectional, and requires configuration on only one of the peers.

SUMMARY STEPS

1. **configure**
2. **l2tp-class** *word*
3. **digest** {**check disable** | **hash** {**MD5** | **SHA1**}] | **secret** {**0** | **7**} *password*]
4. **hidden**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2tp-class** *word* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 3	**digest** {**check disable** | **hash** {**MD5** | **SHA1**}] | **secret** {**0** | **7**} *password*] **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# digest secret cisco hash sha`	Enables L2TPv3 control-channel authentication or integrity checking. • **secret**—Enables L2TPv3 control-channel authentication. **Note** If the **digest** command is issued without the **secret** keyword option, L2TPv3 integrity checking is enabled. • {**0** | **7**}—Specifies the input format of the shared secret. The default value is **0**. – **0**—Specifies that a plain-text secret is entered. – **7**—Specifies that an encrypted secret is entered. • *password*—Defines the shared secret between peer routers. The value entered for the *password* argument must be in the format that matches the input format specified by the {**0** | **7**} keyword option. • **hash** {**MD5** | **SHA1**}—Specifies the hash function to be used in per-message digest calculations. – **MD5**—Specifies HMAC-MD5 hashing (default value). – **SHA1**—Specifies HMAC-SHA-1 hashing.
Step 4	**hidden** **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# hidden`	Enables AVP hiding when sending control messages to an L2TPv3 peer.

Configuring L2TPv3 Digest Secret Graceful Switchover

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

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

L2TPv3 control-channel authentication occurs using a password that is configured on all participating peer PE routers. The L2TPv3 Digest Secret Graceful Switchover feature allows a transition from an old control-channel authentication password to a new control-channel authentication password without disrupting established L2TPv3 tunnels.

Before performing this task, you must enable control-channel authentication (see Configuring L2TPv3 Control Message Hashing, page 7-178).

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

SUMMARY STEPS

1. **configure**
2. **l2tp-class** *word*
3. **digest** {**check disable** | **hash** {**MD5** | **SHA1**}] | **secret** {**0** | **7**} *password*]
4. **end**
   or
   commit
5. **show l2tp tunnel** brief
6. **configure**
7. **l2tp-class** *word*
8. **no digest** [**secret** [**0** | **7**] *password*] [**hash** {**md5** | **sha**}]
9. **end**
   or
   commit
10. **show l2tp tunnel** brief

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2tp-class** *word* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.

	Command or Action	Purpose
Step 3	**digest** {**check disable** | **hash** {**MD5** | **SHA1**}] | **secret** {**0** | **7**} *password*] **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# digest secret cisco hash sha`	Enables L2TPv3 control-channel authentication or integrity checking. • **secret**—Enables L2TPv3 control-channel authentication. **Note** If the **digest** command is issued without the **secret** keyword option, L2TPv3 integrity checking is enabled. • {**0** | **7**}—Specifies the input format of the shared secret. The default value is **0**. – **0**—Specifies that a plain-text secret is entered. – **7**—Specifies that an encrypted secret is entered. • *password*—Defines the shared secret between peer routers. The value entered for the *password* argument must be in the format that matches the input format specified by the {**0** | **7**} keyword option. • **hash** {**MD5** | **SHA1**}—Specifies the hash function to be used in per-message digest calculations. – **MD5**—Specifies HMAC-MD5 hashing (default value). – **SHA1**—Specifies HMAC-SHA-1 hashing.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2tp-class)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2tp-class)# commit`
		Saves configuration changes.
		• When you enter the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

	Command or Action	Purpose
Step 5	**show l2tp tunnel brief** **Example:** `RP/0/RP0/CPU0:router# show l2tun tunnel brief`	Displays the current state of L2 tunnels and information about configured tunnels, including local and remote L2 Tunneling Protocol (L2TP) hostnames, aggregate packet counts, and control-channel information. **Note** Use this command to determine if any tunnels are not using the new password for control-channel authentication. The output displayed for each tunnel in the specified L2TP class should show that two secrets are configured.
Step 6	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 7	**l2tp-class** *word* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 8	**no digest** {**check disable** | **hash** {**MD5** | **SHA1**}] | **secret** {**0** | **7**} *password*] **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# no digest secret cisco hash sha1`	Disables L2TPv3 control-channel authentication or integrity checking.

	Command or Action	Purpose

Step 9	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2tp-class)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2tp-class)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 10	**show l2tp tunnel brief** **Example:** `RP/0/RP0/CPU0:router# show l2tun tunnel brief`	Displays the current state of L2 tunnels and information about configured tunnels, including local and remote L2 Tunneling Protocol (L2TP) hostnames, aggregate packet counts, and control-channel information. Tunnels should no longer be using the old control-channel authentication password. If a tunnel does not update to show that only one secret is configured after several minutes have passed, that tunnel can be manually cleared and a defect report should be filed with TAC. **Note** Issue this command to ensure that all tunnels are using only the new password for control-channel authentication. The output displayed for each tunnel in the specified L2TP class should show that one secret is configured.

Configuring L2TP Control-Channel Maintenance Parameters

Perform this task to configure the interval used for hello messages in an L2TP class.

SUMMARY STEPS

1. **configure**
2. **l2tp-class** *word*
3. **hello** *interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2tp-class** *word* **Example:** `RP/0/RP0/CPU0:router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode.
Step 3	**hello** *interval* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-class)# hello 100`	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 L2TPv3 Pseudowires

Perform the following tasks to configure static and dynamic L2TPv3 pseudowires:

- Configuring a Dynamic L2TPv3 Pseudowire, page 7-184
- Configuring aStatic L2TPv3 Pseudowire, page 7-187

Configuring a Dynamic L2TPv3 Pseudowire

Perform this task to configure a dynamic L2TPv3 pseudowire.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **xconnect group** *name*
4. **p2p** *name*
5. **neighbor** *ip-address* **pw-id** *number*
6. **pw-class** *pw-class-name*
7. **end**
   or
   commit
8. **pw-class** *pw-class-name*
9. **encapsulation l2tpv3**
10. **protocol l2tpv3 class** *class-name*

11. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enter L2VPN configure submode.
Step 3	**xconnect group** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group grp_01`	Enter a name for the cross-connect group.
Step 4	**p2p** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p AC1_to_PW1`	Enters p2p configuration submode to configure point-to-point cross-connects.
Step 5	**neighbor** *ip-address* **pw-id** *number* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.1.1.1 pw-id 665`	Configures a pseudowire for a cross-connect.
Step 6	**pw-class** *pw-class-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class atom`	Enters pseudowire class submode to define a name for the cross-connect.

	Command or Action	Purpose

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 8	**pw-class** *pw-class-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# pw-class class100`	Enters pseudowire class submode to define a pseudowire class template.

Step 9	`encapsulation l2tpv3`
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-pwc)# encapsulation l2tpv3`
		Configures L2TPv3 pseudowire encapsulation.

	Command or Action	Purpose
Step 10	**protocol l2tpv3 class** *class name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-l2tpv3)# protocol l2tpv3 class wkg`	Configures the dynamic pseudowire signaling protocol.

Step 11	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-l2tpv3)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-l2tpv3)# commit`
		Saves configuration changes.
		• When you enter the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• When you enter the **commit** command, the system saves the configuration changes to the running configuration file and remains within the configuration session.

Configuring aStatic L2TPv3 Pseudowire

Perform this task to configure a static L2TPv3 pseudowire.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **xconnect group** *name*
4. **p2p** *name*
5. **neighbor** *ip-address* **pw-id** *number*
6. **l2tp static local session** {*session-id*}
7. **l2tp static local cookie size** {**0** | **4** | **8**} [**value** {*low-value*} [{*high-value*}]]
8. **l2tp static remote session** {*session-id*}
9. **l2tp static remote cookie size** {**0** | **4** | **8**} [**value** {*low-value*} [{*high-value*}]]
10. **pw-class** *name*
11. **end**
    or
    commit
12. **configure**

13. **l2vpn**
14. **pw-class** *name*
15. **encapsulation l2tpv3**
16. ipv4 source *ip-address*
17. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enter L2VPN configure submode.
Step 3	**xconnect group** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group customer_X`	Enter a name for the cross-connect group.
Step 4	**p2p** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p AC1_to_PW1`	Enters p2p configuration submode to configure point-to-point cross-connects.
Step 5	**neighbor** *ip-address* **pw-id** *number* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.1.1.1 pw-id 666`	Configures a pseudowire for a cross-connect.
Step 6	**l2tp static local session** {*session-id*} **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# l2tp static local session 147`	Configures a L2TP pseudowire static session ID.
Step 7	***l2tp static local cookie size*** *{***0** | **4** | **8***}* *[**value** {low-value} [{high-value}]]* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# l2tp static local cookie size 4 value 0XA`	Configures a L2TP pseudowire static session cookie.

	Command or Action	Purpose
Step 8	**l2tp static remote session** {*session-id*} **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# l2tp static remote session 123`	Configures a L2TP pseudowire remote session ID.
Step 9	**l2tp static remote cookie size** {*0* | *4* | *8*} [**value** {*low-value*} [{*high-value*}]] **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# l2tp static remote cookie size 8 value 0x456 0xFFB`	Configures a L2TP pseudowire remote session cookie.
Step 10	**pw-class** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class atom`	Enters pseudowire class submode to define a pseudowire class template.

Step 11	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 12	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 13	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enter L2VPN configure submode.

	Command or Action	Purpose
Step 14	**pw-class** *name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# pw-class class100`	Enters pseudowire class submode to define a pseudowire class template.
Step 15	**encapsulation l2tpv3** **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-pwc)# encapsulation l2tpv3`	Configures L2TPv3 pseudowire encapsulation.
Step 16	**ipv4 source** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-l2tp-pwc-encap-l2tpv3)# ipv4 source 126.10.1.55`	Configures the local source IPv4 address.

Step 17	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-l2vpn-pwc)# end`
	or
	`RP/0/RP0/CPU0:router(config-l2vpn-pwc)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Layer 2 Tunnel Protocol Version 3

This section provides the following configuration examples:

- Configuring an L2TP Class for L2TPv3-based L2VPN PE Routers: Example, page 7-191
- Configuring a Pseudowire Class: Example, page 7-191
- Configuring L2TPv3 Control Channel Parameters: Example, page 7-191
- Configuring an Interface for Layer 2 Transport Mode: Example, page 7-191

Configuring an L2TP Class for L2TPv3-based L2VPN PE Routers: Example

The following example shows how to configure a L2TP class with L2TPv3 based L2VPN for a PE router.

```
configure
  l2tp-class Class_l2tp_01
    receive-window 256
    retransmit retries 8
    retransmit initial retries 10
    retransmit initial timeout max 4
    retransmit initial timeout min 2
    timeout setup 90
    hostname PE1
    hello-interval 100
    digest secret cisco hash MD5
  end
```

Configuring a Pseudowire Class: Example

The following example shows a pseudowire class configuration on a PE router:

```
configure
 l2vpn
  pw-class FR1
   encapsulation l2tpv3
    protocol l2tpv3 [class {class name}]}
    dfbit set
    tos {reflect|value {value}}
    ttl {1-255}
    pmtu max {68-65535}
    ipv4 source {ipv4_address}
    cookie size {0|4|8}
```

Configuring L2TPv3 Control Channel Parameters: Example

The following example shows a typical L2TPv3 control-channel configuration:

```
configure
 l2tp-class FR-l2tp
  authentication
  hostname R2-PE1
  password 7 121A0C041104
  hello-interval 10
  digest secret 7 02050D480809
```

Configuring an Interface for Layer 2 Transport Mode: Example

The following example shows how to configure an interface to operate in Layer 2 transport mode:

```
configure
 interface GigabitEthernet0/4/0/5 l2transport
  negotiation auto

l2vpn
 xconnect group PP-2101
  p2p xc2101
```

```
 interface GigabitEthernet0/4/0/5
 neighbor 150.150.150.250 pw-id 5
  pw-class l2tpv3_class100
 !
!
```

Additional References

The following sections provide additional information related to L2TPv3.

Related Documents

Related Topic	Document Title
MPLS VPN-related commands	*MPLS Virtual Private Network Commands on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Command Reference*
MPLS Layer 2 VPNs	*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*
MPLS Layer 3 VPNs	*Implementing MPLS Layer 3 VPNs on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*
MPLS VPNs over IP Tunnels	*MPLS VPNs over IP Tunnels on Cisco IOS XR Software* module in *Cisco IOS XR MPLS Configuration Guide*
Cisco CRS router getting started material	*Cisco IOS XR Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*

Standards

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

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

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

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport