Configuring Client-Initiated Dial-In VPDN Tunneling

Client-initiated dial-in virtual private dialup networking (VPDN) tunneling deployments allow remote users to access a private network over a shared infrastructure with end-to-end protection of private data. Client-initiated VPDN tunneling does not require additional security to protect data between the client and the Internet service provider (ISP) network access server (NAS).

Finding Feature Information

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

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

Prerequisites for Client-Initiated VPDN Tunneling

- If the client device is a PC, appropriate Virtual Private Network (VPN) software must be installed and configured. For information on installing and configuring client VPN software, refer to the instructions provided with the VPN software package.
- The NAS should be configured to receive incoming calls from clients using ISDN, the public switched telephone network (PSTN), digital subscriber line (DSL), or cable modem.

- The interface between the NAS and the tunnel server must be configured for PPP.
- Before performing the tasks documented in this module, you must perform the required tasks in the Configuring AAA for VPDNs module.

Restrictions for Client-Initiated VPDN Tunneling

- The Layer 2 Forwarding (L2F) protocol is not supported.
- Layer 2 Tunneling Protocol (L2TP) and L2TP Version 3 (L2TPv3) protocols are supported only for tunnels initiated by a client router.
- The Point-to-Point Tunneling Protocol (PPTP) is supported only for tunnels initiated by a client PC running appropriate VPN software.

Information About Client-Initiated VPDN Tunneling

Client-Initiated VPDN Tunneling

Client-initiated dial-in VPDN tunneling is also known as voluntary tunneling. In a client-initiated dial-in VPDN scenario, the client device initiates a Layer 2 tunnel to the tunnel server, and the NAS does not participate in tunnel negotiation or establishment. In this scenario the NAS is not a tunnel endpoint, it simply provides internet connectivity.

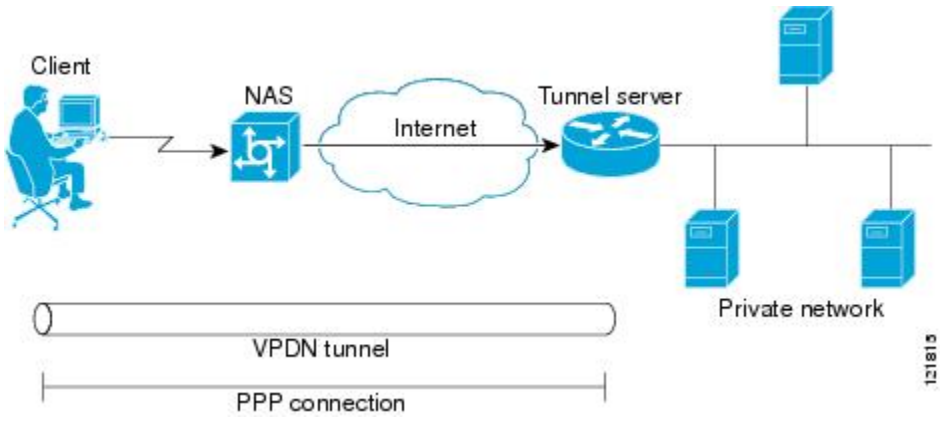
The client can be either of these devices:

- A properly configured router attached to a client network using either L2TP or L2TPv3.
- A PC that is running appropriate VPN client software using PPTP.

Client-initiated VPDN tunneling provides end-to-end security for the connection from the client to the tunnel server. Unlike NAS-initiated VPDN scenarios, no additional security is required to protect the connection between the client device and the NAS.

The figure below depicts a generic client-initiated VPDN tunneling scenario. The local device, which can be either a client PC or a client router, connects to the NAS through a medium that supports PPP. The client can initiate a VPDN tunnel to the tunnel server using either the PPTP, L2TP, or L2TPv3 protocol. The type of Layer 2 tunnel that is established is dependent on the configuration of both the client device and remote tunnel server.

Figure 1: Client-Initiated Tunneling



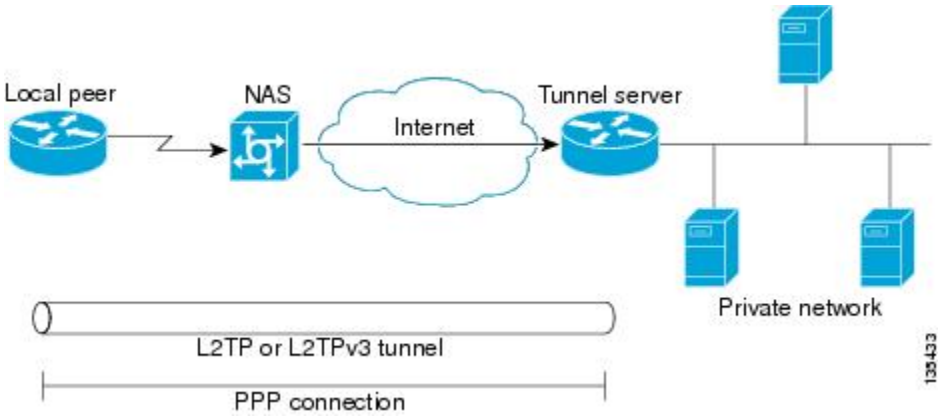
Client-Initiated VPDN Tunneling Using the L2TP or L2TPv3 Protocol

Client-initiated tunnels using the L2TP or L2TPv3 protocol must be initiated by a router configured as the local peer. The L2TP and L2TPv3 protocols are not supported for client-initiated tunnels from a client PC.

In the client-initiated tunneling scenario depicted in the figure below, the local peer connects to the NAS through a medium that supports PPP, such as a dialup modem, DSL, ISDN, or cable modem. The PPP interface adds Layer 2 encapsulation to Layer 3 packets, allowing them to be sent to the tunnel server over an L2TP or L2TPv3 tunnel.

The client can initiate a VPDN tunnel to the tunnel server using either the L2TP or L2TPv3 protocol. The type of Layer 2 tunnel that is established is dependent on the configuration of both the local peer and remote tunnel server. The local and remote peers must be configured to establish the same type of tunnel.

Figure 2: L2TP or L2TPv3 Client-Initiated Tunneling

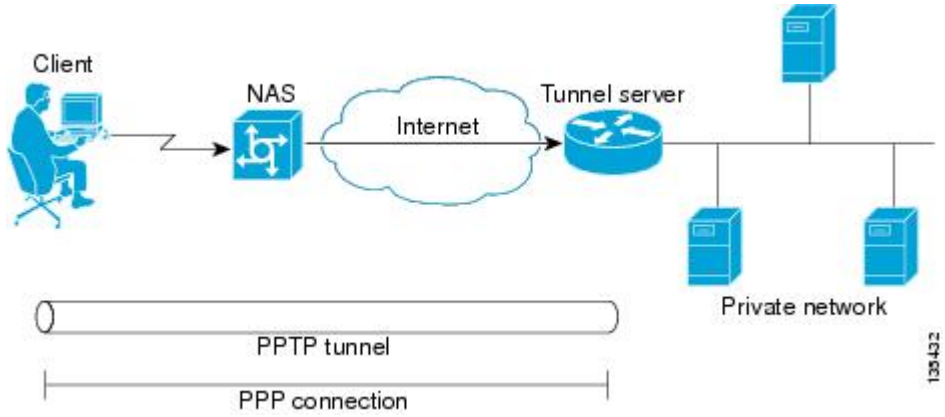


Client-Initiated VPDN Tunneling Using the PPTP Protocol

Client-initiated tunnels using the PPTP protocol must be initiated by a client PC configured with appropriate VPN client software. The client must manage the software that initiates the tunnel on the PC. The PPTP protocol is not supported for client-initiated tunnels from a local peer router.

In the client-initiated tunneling scenario depicted in the figure below, the client PC connects to the NAS through a medium that supports PPP, such as a dialup modem, DSL, ISDN, or cable modem. The client can initiate a VPDN tunnel to the tunnel server using the PPTP protocol.

Figure 3: PPTP Client-Initiated Tunneling



PPTP uses an enhanced Generic Routing Encapsulation (GRE) mechanism to provide a flow- and congestion-controlled encapsulated datagram service for carrying PPP packets.

These sections contain information about PPTP features:

MPPE Encryption of PPTP Tunnels

Microsoft Point-to-Point Encryption (MPPE) can be used to encrypt PPTP VPDN tunnels. MPPE encrypts the entire session from the client to the tunnel server.

MPPE is an encryption technology developed by Microsoft to encrypt point-to-point links. These connections can be over a dialup line or over a VPDN tunnel. MPPE works is a feature of Microsoft Point-to-Point Compression (MPPC).

MPPC is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections.

MPPE is negotiated using bits in the MPPC option within the Compression Control Protocol (CCP) MPPC configuration option (CCP configuration option number 18).

MPPE uses the RC4 algorithm with either 40- or 128-bit keys. All keys are derived from the cleartext authentication password of the user. RC4 is stream cipher; therefore, the sizes of the encrypted and decrypted frames are the same size as the original frame. The Cisco implementation of MPPE is fully interoperable with that of Microsoft and uses all available options, including stateless mode (sometimes referred to as historyless mode). Stateless mode can increase throughput in lossy environments such as VPDNs, because neither side needs to send CCP Resets Requests to synchronize encryption contexts when packets are lost.

Two modes of MPPE encryption are available:

- Stateful MPPE encryption--Stateful encryption provides the best performance but might be adversely affected by networks that experience substantial packet loss. Because of the way that the RC4 tables are reinitialized during stateful synchronization, it is possible that two packets might be encrypted using the same key. For this reason, stateful encryption might not be appropriate for lossy network environments (such as Layer 2 tunnels on the Internet). If you configure stateful encryption, the PPTP flow control alarm is automatically enabled.

- Stateless MPPE encryption--Stateless encryption provides a lower level of performance, but will be more reliable in a lossy network environment. Stateless mode is sometimes referred to as historyless mode. The PPTP flow control alarm is automatically disabled when stateless encryption is being used.

PPTP Flow Control Alarm

The PPTP flow control alarm indicates when congestion or lost packets are detected. When the flow control alarm goes off, PPTP reduces volatility and additional control traffic by falling back from a stateful to a stateless encryption mode for the MPPE session.

How to Configure Client-Initiated VPDN Tunneling

Configuring Client-Initiated Tunneling Using the L2TP or L2TPv3 Protocol

Prerequisites

- This procedure requires Cisco IOS Release 12.3(2)T or a later release on both the local peer and the tunnel server for L2TPv3 tunneling configurations.
- This procedure requires Cisco IOS Release 12.3(2)T or a later release on the local peer for L2TP tunneling configurations.
- Cisco Express Forwarding must be enabled.

Restrictions

- PPP is the only encapsulation method supported.
- PPTP tunneling is not supported.
- Session establishment cannot be triggered by interesting traffic.
- Failover is not supported with the L2TP peer.
- L2TP redirect is not supported.

Configuring L2TP or L2TPv3 Client-Initiated VPDN Tunneling on the Local Peer

Perform this task to configure the local peer to initiate VPDN tunnels to the tunnel server. This task applies to both L2TP and L2TPv3 configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **exit**
5. **pseudowire-class** [*pw-class-name*]
6. **exit**

7. **interface virtual-ppp** *number*
8. **ip unnumbered** *interface-type interface-number*
9. **ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
10. **ppp chap hostname** [*hostname*]
11. **pseudowire** *peer-ip-address vcid* **pw-class** *pw-class-name* [**sequencing** {**transmit** | **receive** | **both**}]
12. **exit**
13. **ip route** *prefix mask* {*ip-address*| *interface-type interface-number* [*ip-address*]} [**distance**] [**name**] [**permanent**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Router(config)# l2tp-class l2tpclass2`	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. • You can configure L2TP control channel parameters in L2TP class configuration mode. See the Configuring L2TP Control Channel Parameters, on page 13 for more information.
Step 4	**exit** **Example:** `Router(config-l2tp-class)# exit`	Exits L2TP class configuration mode.
Step 5	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class pwclass2`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
Step 6	**exit** **Example:** `Router(config-pw)# exit`	Exits pseudowire class configuration mode.

	Command or Action	Purpose
Step 7	**interface virtual-ppp** *number* **Example:** `Router(config)# interface virtual-ppp 2`	Enters interface configuration mode and assigns a virtual-PPP interface number.
Step 8	**ip unnumbered** *interface-type interface-number* **Example:** `Router(config-if)# ip unnumbered loopback 1`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 9	**ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-if)# ppp authentication chap`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication is selected on the interface.
Step 10	**ppp chap hostname** [*hostname*] **Example:** `Router(config-if)# ppp chap hostname peer2`	Creates a pool of dialup routers that all appear to be the same host when authenticating with CHAP.
Step 11	**pseudowire** *peer-ip-address vcid* **pw-class** *pw-class-name* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# pseudowire 172.16.32.24 10 pw-class pwclass2`	Specifies the IP address of the tunnel server and the 32-bit virtual circuit identifier (VCID) shared between the devices at each end of the control channel. • *peer-ip-address vcid* --The tunnel server IP address and VCID must be a unique combination on the router. **Note** For L2TPv3 tunnels, the VCID configured on the local peer must match the VCID configured on the tunnel server. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type will be taken.The **pw-class** keyword binds the pseudowire statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. • **sequencing** --The optional **sequencing** keyword specifies whether sequencing is required for packets that are received, sent, or both received and sent. **Note** If the network between the tunnel endpoints is unreliable, packets might be delivered out of order. Enabling sequencing can reduce the number of dropped packets and network latency.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 13	**ip route** *prefix mask* {*ip-address*| *interface-type interface-number* [*ip-address*]} [**distance**] [**name**] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.0 255.255.255.0 virtual-PPP 1`	Establishes static routes.

What to Do Next

You must perform one of these tasks depending on the tunneling protocol you are configuring:

- Configuring Client-Initiated Tunneling on the Tunnel Server for L2TP Tunnels
- Configuring Client-Initiated Tunneling on the Tunnel Server for L2TPv3 Tunnels

Configuring Client-Initiated Tunneling on the Tunnel Server for L2TP Tunnels

When a request to establish an L2TP tunnel is received by the tunnel server, the tunnel server must create a virtual access interface. The virtual access interface is cloned from a virtual template interface, used, and then freed when no longer needed. The virtual template interface is a logical entity that is not tied to any physical interface. The tunnel server must be configured to terminate VPDN tunnels.

Perform this task to configure the tunnel server to terminate client-initiated L2TP tunnels and to configure a basic virtual template.

Before you begin

- You must perform the required tasks in the Configuring AAA for VPDNs module.
- The same tunneling protocol must be configured on the tunnel server and the local peer device. For L2TP tunnels, the tunneling protocol is configured in a VPDN group on the tunnel server. On the local peer, the tunneling protocol is configured in a pseudowire class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn-group** *name*
4. **description** *string*
5. **accept-dialin**
6. **protocol l2tp**
7. **virtual-template** *template-number*
8. **exit**

9. **terminate-from hostname** *hostname*
10. **exit**
11. **interface virtual-template** *number*
12. **ip unnumbered** *interface-type interface-number*
13. **ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
14. **ppp chap hostname** [*hostname*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn-group** *name* **Example:** `Router(config)# vpdn group vpdngroup1`	Enters VPDN group configuration mode and associates a VPDN group to a customer or VPDN profile.
Step 4	**description** *string* **Example:** `Router(config-vpdn)# description clientl2tp`	(Optional) Adds a description to a VPDN group.
Step 5	**accept-dialin** **Example:** `Router(config-vpdn)# accept-dialin`	Enters VPDN accept-dialin configuration mode, configures the tunnel server to accept tunneled PPP connections, and creates an accept-dialin VPDN subgroup.
Step 6	**protocol l2tp** **Example:** `Router(config-vpdn-acc-in)# protocol l2tp`	Specifies the Layer 2 protocol that the VPDN subgroup will use.
Step 7	**virtual-template** *template-number* **Example:** `Router(config-vpdn-acc-in)# virtual-template 1`	Specifies which virtual template will be used to clone virtual access interfaces.
Step 8	**exit** **Example:**	Exits VPDN accept-dialin configuration mode.

	Command or Action	Purpose
	`Router(config-vpdn-acc-in)# exit`	
Step 9	**terminate-from hostname** *hostname* **Example:** `Router(config-vpdn)# terminate-from hostname peer1`	Specifies the hostname of the remote LAC or LNS that will be required when accepting a VPDN tunnel.
Step 10	**exit** **Example:** `Router(config-vpdn)# exit`	Exits VPDN group configuration mode.
Step 11	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Enters interface configuration mode and creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
Step 12	**ip unnumbered** *interface-type interface-number* **Example:** `Router(config-if)# ip unnumbered loopback 1`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 13	**ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-if)# ppp authentication chap`	Enables CHAP or PAP or both and specifies the order in which CHAP and PAP authentication is selected on the interface.
Step 14	**ppp chap hostname** [*hostname*] **Example:** `Router(config-if)# ppp chap hostname peer2`	Creates a pool of dialup routers that all appear to be the same host when authenticating with CHAP.

What to Do Next

You must perform the task in the Configuring the Pseudowire, on page 18.

Configuring Client-Initiated Tunneling on the Tunnel Server for L2TPv3 Tunnels

The tunnel server must be configured to terminate VPDN tunnels. The same tunneling protocol must be configured on the tunnel server and the local peer device. For L2TPv3 tunnels, the tunneling protocol is configured in a pseudowire class on both the tunnel server and the local peer.

Perform this task to configure the tunnel server to terminate client-initiated L2TPv3 tunnels.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **exit**
5. **pseudowire-class** [*pw-class-name*]
6. **exit**
7. **interface virtual-ppp** *number*
8. **ip unnumbered** *interface-type interface-number*
9. **ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
10. **ppp chap hostname** [*hostname*]
11. **pseudowire** *peer-ip-address vcid* **pw-class** *pw-class-name* [**sequencing** {**transmit** | **receive** | **both**}]
12. **exit**
13. **ip route** *prefix mask* {*ip-address*| *interface-type interface-number* [*ip-address*]} [**distance**] [**name**] [**permanent**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Router(config)# l2tp-class l2tpclass2`	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. • You can configure L2TP control channel parameters in L2TP class configuration mode. See the Configuring L2TP Control Channel Parameters, on page 13.
Step 4	**exit** **Example:** `Router(config-l2tp-class)# exit`	Exits L2TP class configuration mode.
Step 5	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class pwclass2`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-pw)# exit`	Exits pseudowire class configuration mode.
Step 7	**interface virtual-ppp** *number* **Example:** `Router(config)# interface virtual-ppp 2`	Enters interface configuration mode and assigns a virtual-PPP interface number.
Step 8	**ip unnumbered** *interface-type interface-number* **Example:** `Router(config-if)# ip unnumbered loopback 1`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 9	**ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-if)# ppp authentication chap`	Enables CHAP or PAP or both and specifies the order in which CHAP and PAP authentication is selected on the interface.
Step 10	**ppp chap hostname** [*hostname*] **Example:** `Router(config-if)# ppp chap hostname peer2`	Creates a pool of dialup routers that all appear to be the same host when authenticating with CHAP.
Step 11	**pseudowire** *peer-ip-address vcid* **pw-class** *pw-class-name* [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# pseudowire 172.16.32.24 10 pw-class pwclass2`	Specifies the IP address of the local peer and the 32-bit VCID shared between the local peer and the tunnel server. • *peer-ip-address vcid* --The peer router IP address and VCID must be a unique combination on the router. **Note** The VCID configured on the tunnel server must match the VCID configured on the local peer. • **pw-class** *pw-class-name* --The pseudowire class configuration from which the data encapsulation type will be taken.The **pw-class** keyword binds the pseudowire statement to a specific pseudowire class. The pseudowire class then serves as the template configuration for all attachment circuits bound to it. • **sequencing** --The optional **sequencing** keyword specifies whether sequencing is required for packets that are received, sent, or both received and sent.

	Command or Action	Purpose
		Note If the network between the tunnel endpoints is unreliable, packets might be delivered out of order. Enabling sequencing can reduce the number of dropped packets and network latency.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 13	**ip route** *prefix mask* {*ip-address*| *interface-type interface-number* [*ip-address*]} [**distance**] [**name**] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.0 255.255.255.0 Virtual-PPP 1`	Establishes static routes.

What to Do Next

You must perform the task in the Configuring the Pseudowire, on page 18.

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. Configuring L2TP control channel parameters is optional.

The three groups of L2TP control channel parameters that you can configure for an L2TP class are described in these sections:

After the router enters 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.

Prerequisites

L2TP Tunnels

For L2TP, the L2TP class is configured only on the local peer. An L2TP class was defined for the local peer in the Configuring L2TP or L2TPv3 Client-Initiated VPDN Tunneling on the Local Peer, on page 5."

L2TPv3 Tunnels

For L2TPv3, an L2TP class must be configured on both the local peer and the tunnel server. An L2TP class was defined for the local peer in the Configuring L2TP or L2TPv3 Client-Initiated VPDN Tunneling on the Local Peer, on page 5. An L2TP class was defined for the tunnel server in the Configuring Client-Initiated Tunneling on the Tunnel Server for L2TPv3 Tunnels, on page 10.

Configuring L2TP Control Channel Timing Parameters

These 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

Perform this task to configure a set of timing control channel parameters for an L2TP class. All of the timing control channel parameter configurations are optional and can be configured in any order. If these parameters are not configured, the default values are applied.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name*for each one.
Step 4	**receive-window** *size* **Example:** `Router(config-l2tp-class)# receive-window 30`	(Optional) Configures the number of packets that can be received by the remote peer before backoff queueing occurs. • The valid values range from 1 to the upper limit the peer has for receiving packets. The default value is the upper limit.
Step 5	**retransmit** {**initial retries** *initial-retries* | **retries** *retries*| **timeout** {**max** | **min**} *timeout*}	(Optional) Configures parameters that affect the retransmission of control packets.

	Command or Action	Purpose
	Example: `Router(config-l2tp-class)# retransmit retries 10`	• **initial retries** --Specifies how many start control channel requests (SCCRQs) are re-sent before the device gives 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 the device determines that the peer 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 6	**timeout setup** *seconds*
	Example:
	`Router(config-l2tp-class)#`
	`timeout setup 400`
		(Optional) Configures the amount of time, in seconds, allowed for setting up a control channel.
		• Valid values for the *seconds* argument range from 60 to 6000. The default value is 300.

What to Do Next

You must perform the task in the Configuring the Pseudowire, on page 18.

Configuring L2TP Control Channel Authentication Parameters

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

- Authentication for the L2TP control channel
- Local hostname used for authenticating the control channel
- Hiding the attribute-value (AV) pairs in outgoing control messages
- Password used for control channel authentication and AV pair hiding

Perform this task to configure a set of authentication control channel parameters for an L2TP class. All of the authentication control channel parameter configurations are optional and can be configured in any order. If these parameters are not configured, the default values will be applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2tp-class** [*l2tp-class-name*]
4. **authentication**
5. **hostname** *name*
6. **hidden**

7. **password** [*encryption-type*] *password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2tp-class** [*l2tp-class-name*] **Example:** `Router(config)# l2tp-class class1`	Specifies the L2TP class name and enters L2TP class configuration mode. • The *l2tp-class-name* argument is optional. However, if you want to configure multiple L2TP classes you must specify a unique *l2tp-class-name*for each one.
Step 4	**authentication** **Example:** `Router(config-l2tp-class)# authentication`	(Optional) Enables authentication for the control channel between PE routers. • Authentication is enabled by default.
Step 5	**hostname** *name* **Example:** `Router(config-l2tp-class)# hostname yb2`	(Optional) 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.

Step 6	**hidden**
	Example:
	`Router(config-l2tp-class)#`
	`hidden`
		(Optional) Hides the AV pairs in control messages.
		• AV pairs are not hidden by default.

Step 7	**password** [*encryption-type*] *password*
	Example:
	`Router(config-l2tp-class)#`
	`password tunnel2`
		(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.

	Command or Action	Purpose
		Note The password configured on the local peer must match the password configured on the tunnel server.

What to Do Next

You must perform the task in the Configuring the Pseudowire, on page 18.

Configuring L2TP Control Channel Maintenance Parameters

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

Perform this task to configure the interval used for hello messages for an L2TP class. This control channel parameter configuration is optional. If this parameter is not configured, the default value will be applied.

SUMMARY STEPS

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

DETAILED STEPS

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

Step 4	**hello** *interval*
	Example:
	`Router(config-l2tp-class)#`
	`hello 100`
		(Optional) Specifies the exchange interval (in seconds) used between L2TP hello packets.
		• Valid values for the *interval* argument range from 0 to 1000. The default value is 60.

What to Do Next

You must perform the task in the Configuring the Pseudowire, on page 18.

Configuring the 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 L2TP or L2TPv3 sessions that will be used to transport attachment circuit traffic over the pseudowire.

The pseudowire configuration specifies the characteristics of the L2TP or L2TPv3 signaling mechanism, including the data encapsulation type, the control protocol, sequencing, fragmentation, payload-specific options, and IP properties. The setting that determines if signaling is used to set up the pseudowire is also included.

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. This configuration could prevent a control channel from being established.

If you do not configure the optional pseudowire class configuration commands, the default values are used.

Before you begin

L2TP Tunnels

For L2TP, the pseudowire class is configured only on the local peer. A pseudowire class was defined for the local peer in the task Configuring L2TP or L2TPv3 Client-Initiated VPDN Tunneling on the Local Peer, on page 5.

L2TPv3 Tunnels

For L2TPv3, the pseudowire class must be configured on both the local peer and the tunnel server. A pseudowire class was defined for the local peer in the task Configuring L2TP or L2TPv3 Client-Initiated VPDN Tunneling on the Local Peer, on page 5. A pseudowire class was defined for the tunnel server in the task Configuring Client-Initiated Tunneling on the Tunnel Server for L2TPv3 Tunnels, on page 10.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation** {**l2tpv2** | **l2tpv3**}
5. **protocol** {**l2tpv2** | **l2tpv3**} [*l2tp-class-name*]
6. **ip local interface** *interface-name*
7. **ip pmtu**
8. **ip tos** {**value** *value* | **reflect**}
9. **ip dfbit set**
10. **ip ttl** *value*
11. **sequencing** {**transmit** | **receive** | **both**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class etherpw`	Enters pseudowire class configuration mode and optionally specifies the name of the L2TP pseudowire class.
Step 4	**encapsulation** {**l2tpv2** | **l2tpv3**} **Example:** `Router(config-pw)# encapsulation l2tpv3`	Specifies the data encapsulation method used to tunnel IP traffic. • **l2tpv2** --L2TP is the tunneling method to be used to encapsulate data in the pseudowire. • **l2tpv3** --L2TPv3 is the tunneling method to be used to encapsulate data in the pseudowire.

Step 5	**protocol** {**l2tpv2** | **l2tpv3**} [*l2tp-class-name*]
	Example:
	`Router(config-pw)#`
	`protocol l2tpv3 class1`
		Specifies the Layer 2 signaling protocol to be used to manage the pseudowires created with the control channel parameters in the specified L2TP class.
		• **l2tpv2** --Specifies L2TP as the signaling protocol to be used.
		• **l2tpv3** --Specifies L2TPv3 as the signaling protocol to be used.
		• *l2tp-class-name* --(Optional) The name of the L2TP class configuration to be used for pseudowires set up from the pseudowire class.
		Note If the *l2tp-class-name* argument is not specified, the default values for L2TP control channel parameters will be used.

Step 6	**ip local interface** *interface-name*
	Example:
	`Router(config-pw)#`
	`ip local interface e0/0`
		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 or Action	Purpose
		Note This command must be configured for pseudowire class configurations using L2TPv3 as the data encapsulation method.

Step 7	**ip pmtu**
	Example:
	`Router(config-pw)#`
	`ip pmtu`
		(Optional) Enables the discovery of the path maximum transmission unit (PMTU) for tunneled traffic.
		• This command enables the processing of Internet Control Message Protocol (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 Don't Fragment (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.
		• This command must be enabled in the pseudowire class configuration for fragmentation of IP packets before the data enters the pseudowire to occur.
		Note For fragmentation of IP packets before the data enters the pseudowire, we recommend that you also enable the **ip dfbit set** command in the pseudowire class configuration. This allows the PMTU to be obtained more rapidly.

Step 8	**ip tos** {**value** *value* | **reflect**} **Example:** `Router(config-pw)# ip tos reflect`	(Optional) Configures the value of the type of service (ToS) byte in IP headers of tunneled packets, or reflects the ToS byte value from the inner IP header. • Valid values for the *value* argument range from 0 to 255. The default ToS byte value is 0.
Step 9	**ip dfbit set** **Example:** `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 10	**ip ttl** *value* **Example:** `Router(config-pw)# ip ttl 100`	(Optional) Configures the value of the time to live (TTL) byte in the IP headers of tunneled packets. • Valid values for the *value* argument range from 1 to 255. The default TTL byte value is 255.
Step 11	**sequencing** {**transmit** | **receive** | **both**} **Example:**	(Optional) Specifies the direction in which sequencing of data packets in a pseudowire is enabled.

	Command or Action	Purpose
	`Router(config-pw)# sequencing both`	• **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. **Note** If the network between the tunnel endpoints is unreliable, packets might be delivered out of order. Enabling sequencing can reduce the number of dropped packets and network latency.

Verifying an L2TP Control Channel

Perform this task to display detailed information about the L2TP control channels that are set up to other L2TP-enabled devices for all L2TP sessions on the router.

SUMMARY STEPS

1. **enable**
2. **show l2tun tunnel all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show l2tun tunnel all** **Example:** `Router# show l2tun tunnel all`	Displays the current state of Layer 2 tunnels and information about configured tunnels, including local and remote L2TP hostnames, aggregate packet counts, and control channel information.

Configuring Client-Initiated VPDN Tunneling Using the PPTP Protocol

Prerequisites for Configuring Client-Initiated VPDN Tunneling Using the PPTP Protocol

The client PC must be configured with appropriate VPN client software.

Restrictions for Configuring Client-Initiated VPDN Tunneling Using the PPTP Protocol

- Only Cisco Express Forwarding and process switching are supported. Regular fast switching is not supported.
- PPTP does not support multilink.
- VPDN multihop is not supported.
- Because all PPTP signaling is over TCP, TCP configurations will affect PPTP performance in large-scale environments.
- MPPE is not supported with TACACS.
- Windows clients must use Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) authentication in order for MPPE to work.
- If you are performing mutual authentication with MS-CHAP and MPPE, both sides of the tunnel must use the same password.
- To use MPPE with authentication, authorization, and accounting (AAA), you must use a RADIUS server that supports the Microsoft vendor specific attribute for MPPE-KEYS. CiscoSecure NT supports MPPE beginning with release 2.6. CiscoSecure UNIX does not support MPPE.

Configuring the Tunnel Server to Accept PPTP Tunnels

The tunnel server must be configured to terminate PPTP tunnels.

Perform this task to configure the tunnel server to accept tunneled PPPTP connections from a client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn-group** *name*
4. **accept-dialin**
5. **protocol pptp**
6. **virtual-template** *template-number*

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**vpdn-group** *name* **Example:** `Router(config)# vpdn-group 1`	Creates a VPDN group or associates a VPDN group to a customer or VPDN profile and enters VPDN group configuration mode.
Step 4	**accept-dialin** **Example:** `Router(config-vpdn)# accept-dialin`	Creates an accept dial-in VPDN subgroup that configures a tunnel server to accept requests from a NAS to tunnel dial-in calls, and enters accept dial-in VPDN subgroup configuration mode.
Step 5	**protocol pptp** **Example:** `Router(config-vpdn-acc-in)# protocol pptp`	Specifies the Layer 2 protocol that the VPDN group will use.
Step 6	**virtual-template** *template-number* **Example:** `Router(config-vpdn-acc-in)# virtual-template 1`	Specifies which virtual template will be used to clone virtual access interfaces.

What to Do Next

You must perform the task in the Configuring the Virtual Template on the Tunnel Server, on page 23.

Configuring the Virtual Template on the Tunnel Server

When a request to establish a tunnel is received by the tunnel server, the tunnel server must create a virtual access interface. The virtual access interface is cloned from a virtual template interface, used, and then freed when no longer needed. The virtual template interface is a logical entity that is not tied to any physical interface.

Perform this task on the tunnel server to configure a basic virtual template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ip unnumbered** *type number*
5. **ppp authentication** *protocol1* [*protocol2...*] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**]
6. **peer default ip address** {*ip-address*| **dhcp-pool** | **dhcp** | **pool** [*pool-name*]}
7. **encapsulation** *encapsulation-type*
8. **ppp encrypt mppe** {**auto** | **40** | **128**} [**passive** | **required**] [**stateful**]

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Enters interface configuration mode and creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered FastEthernet 0/0`	Enables IP processing on a serial interface without assigning an explicit IP address to the interface. **Note** Configuring the **ip address** command within a virtual template is not recommended. Configuring a specific IP address in a virtual template can result in the establishment of erroneous routes and the loss of IP packets.
Step 5	**ppp authentication** *protocol1* [*protocol2*...] [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**] **Example:** `Router(config-if)# ppp authentication chap`	Enables at least one PPP authentication protocol and specifies the order in which the protocols are selected on the interface.
Step 6	**peer default ip address** {*ip-address*| **dhcp-pool** | **dhcp** | **pool** [*pool-name*]} **Example:** `Router(config-if)# peer default ip address pool mypool`	Specifies an IP address, an address from a specific IP address pool, or an address from the Dynamic Host Configuration Protocol (DHCP) mechanism to be returned to a remote peer connecting to this interface.
Step 7	**encapsulation** *encapsulation-type* **Example:** `Router(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface.
Step 8	**ppp encrypt mppe** {**auto** | **40** | **128**} [**passive** | **required**] [**stateful**] **Example:** `Router(config-if)# ppp encrypt mppe auto required`	(Optional) Enable MPPE encryption on the virtual template. • **passive** --The tunnel server will not offer MPPE encryption, but will negotiate if the other tunnel endpoint requests encryption. • **required** --MPPE must be negotiated, or the connection will be terminated.

	Command or Action	Purpose
		• **stateful** --MPPE will negotiate only stateful encryption. If the **stateful** keyword is not used, MPPE will first attempt to negotiate stateless encryption, but will allow stateful encryption if the other tunnel endpoint requests the stateful mode.

Configuring MPPE on the ISA Card

Using the Industry-Standard Architecture (ISA) card to offload MPPE from the Route Processor will improve performance in large-scale environments.

Perform this optional task to offload MPPE encryption from the tunnel server processor to the ISA card.

Note An ISA card must be installed on the tunnel server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller isa** *slot* / *port*
4. **encryption mppe**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller isa** *slot* / *port* **Example:** `Router(config)# controller isa 5/0`	Enters controller configuration mode on the ISA card.
Step 4	**encryption mppe** **Example:** `Router(config-controller)# encryption mppe`	Enables MPPE encryption on an ISA card.

What to Do Next

You must reboot your router for the configuration of the **encryption mppe** command to take effect.

Tuning PPTP

You can configure PPTP control options to tune the performance of your PPTP deployment. All of the PPTP tuning configuration commands are optional and can be configured in any order. If these parameters are not configured, the default values are applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn-group** *name*
4. **pptp flow-control receive-window** *packets*
5. **pptp flow-control static-rtt** *timeout-interval*
6. **pptp tunnel echo** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn-group** *name* **Example:** `Router(config)# vpdn group pptp1`	Enters VPDN group configuration mode and associates a VPDN group to a customer or VPDN profile.
Step 4	**pptp flow-control receive-window** *packets* **Example:** `Router(config-vpdn)# pptp flow-control receive-window 20`	Specifies how many packets the client can send before it must wait for the acknowledgment from the tunnel server.
Step 5	**pptp flow-control static-rtt** *timeout-interval* **Example:** `Router(config-vpdn)# pptp flow-control static-rtt 2000`	Specifies the timeout interval of the tunnel server between sending a packet to the client and receiving a response. **Note** If the configured timeout interval elapses with no response, the flow control alarm will be triggered.

	Command or Action	Purpose
Step 6	**pptp tunnel echo** *seconds* **Example:** `Router(config-vpdn)# pptp tunnel echo 90`	Specifies the period of idle time on the tunnel that will trigger an echo message from the tunnel server to the client.

Verifying a PPTP Client-Initiated VPDN Configuration

Perform this task to verify that a PPTP client-initiated VPDN configuration functions properly.

SUMMARY STEPS

1. Dial in to the NAS from a client PC.
2. From the client PC, establish a PPTP connection to the tunnel server using the VPN client software.
3. From the client, ping the remote network.
4. **enable**
5. **show vpdn**
6. **show vpdn session all**
7. **show ppp mppe virtual-access** *number*

DETAILED STEPS

Step 1 Dial in to the NAS from a client PC.

Ensure that the client PC is able to connect to the NAS by establishing a dial-in connection. As the call comes in to the NAS, a LINK-3-UPDOWN message automatically appears on the NAS terminal screen. In the following example, the call comes into the NAS on asynchronous interface 14:

Example:

```
*Jan  1 21:22:18.410: %LINK-3-UPDOWN: Interface Async14, changed state to up
```

Note No **debug** commands are turned on to display this log message. This message should be displayed within 30 seconds after the client first sends the call.

If this message is not displayed by the NAS, there is a problem with the dial-in configuration.

Step 2 From the client PC, establish a PPTP connection to the tunnel server using the VPN client software.

Step 3 From the client, ping the remote network.

From the client desktop:

a) Click **Start**.
b) Choose **Run**.
c) Enter **ping** *remote-ip-address* .
d) Click **OK**.
e) Look at the terminal screen and verify that the remote network is sending ping reply packets to the client.

Step 4 **enable**

Enter this command on the tunnel server to enter privileged EXEC mode. Enter your password if prompted:

Example:

```
Router> enable
```

Step 5 **show vpdn**

Enter this command on the tunnel server to display information about active tunnels and message identifiers. Verify that the client has established a PPTP session.

Example:

```
Router# show vpdn
% No active L2TP tunnels
% No active L2F tunnels
PPTP Tunnel and Session Information (Total tunnels=1 sessions=1)
LocID RemID Remote Name     State    Remote Address  Port  Sessions
13    13    10.1.2.41       estabd   10.1.2.41       1136  1
LocID RemID TunID Intf    Username      State   Last Chg
13    0     13    Vi3                   estabd  000030
```

Step 6 **show vpdn session all**

Enter this command for more detailed information about the VPDN session. The last line of output from the **show vpdn session all** command indicates the current status of the flow control alarm.

Example:

```
Router# show vpdn session all
% No active L2TP tunnels
% No active L2F tunnels
PPTP Session Information (Total tunnels=1 sessions=1)
Call id 13 is up on tunnel id 13
Remote tunnel name is 10.1.2.41
 Internet Address is 10.1.2.41
 Session username is unknown, state is estabd
 Time since change 000106, interface Vi3
 Remote call id is 0
 10 packets sent, 10 received, 332 bytes sent, 448 received
 Ss 11, Sr 10, Remote Nr 10, peer RWS 16
 0 out of order packets
 Flow alarm is clear.
```

Step 7 **show ppp mppe virtual-access** *number*

Enter this command to display MPPE information for the virtual access interface:

Example:

```
Router# show ppp mppe virtual-access 3
Interface Virtual-Access3 (current connection)
  Hardware (ISA5/1, flow_id=13) encryption, 40 bit encryption, Stateless mode
  packets encrypted = 0        packets decrypted  = 1
  sent CCP resets   = 0        receive CCP resets = 0
  next tx coherency = 0        next rx coherency  = 0
  tx key changes    = 0        rx key changes     = 0
  rx pkt dropped    = 0        rx out of order pkt= 0
  rx missed packets = 0
```

To display changed information, reissue the command:

Example:

```
Router# show ppp mppe virtual-access 3
Interface Virtual-Access3 (current connection)
  Hardware (ISA5/1, flow_id=13) encryption, 40 bit encryption, Stateless mode
  packets encrypted = 0        packets decrypted  = 1
  sent CCP resets   = 0        receive CCP resets = 0
  next tx coherency = 0        next rx coherency  = 0
  tx key changes    = 0        rx key changes     = 1
  rx pkt dropped    = 0        rx out of order pkt= 0
```

rx missed packets = 0

Configuring IPv6 Support over Virtual-PPP and L2TPv2

Prerequisites for Configuring IPv6 Support over Virtual-PPP and L2TPv2

- IPv6 over PPP must be supported and functioning.
- IPv4 functionality on Virtual-PPP must be available. This is so that the dual stack functionality is supported.

Restrictions for Configuring IPv6 Support over Virtual-PPP and L2TPv2

- This feature is not applicable to L2TPv3.
- This enhancement will only work on small Cisco ISR boxes like TSN platforms Cisco C1111, Cisco ISR 1100 and Cisco ISR 4221, used with client configurations. It does not work on bigger platforms like Cisco ASR 1000 or Cisco ISR 4450.
- Backup LNS (L2TP Network Server) support is not available.

Information on Configuring IPv6 Support over Virtual-PPP and L2TPv2

Virtual-PPP (Virtual Point-to-Point Protocol) is used along with L2TPv2 tunnel to connect to the LNS (L2TP Network Server). Up until now this support was only for IPv4 packets. Starting from release 17.3.1, this feature enables the support of IPv6 over Virtual-PPP and L2TPv2. It also enables dual-stack support (flow of both IPv4 and IPv6 packets) over Virtual-PPP.

Steps to configure Virtual-PPP client session

1. To create a Virtual-PPP, first create a L2TP session.
2. Create the L2TP tunnel between the LNS and the VPP (Vector Packet Processing) box.
3. Once you have succesfully completed the tunnel establishment, the L2TP session is created.
4. Using this L2TP tunnel, the PPP (Point-to-Point Protocol) starts its negotiation process.
5. First step in the negotiation process is negotiating PPP LCP (Link Control Protocol).
6. Next step is the authentication using either CHAP (Challenge-Handshake Authentication Protocol) or PAP (Password Authentication Protocol).
7. After a succesful authentication, PPP moves to the IPCP (Internet Protocol Control Protocol) phase.

8. In this phase, you can negotiate both IPv4 and IPv6 addresses.
9. Once you finish IP negotiation, the Virtual-PPP interface acquires both IPv4 and IPv6 addresses.
10. The configuration is complete. Traffic flowing through the Virtual-PPP interface will now be attached with L2TP tunnel, UDP (User Datagram Protocol) and PPP headers.

Configuration Examples for Client-Initiated VPDN Tunneling

Example Configuring L2TP Client-Initiated Tunneling

The following example configures L2TP client-initiated tunneling on the local peer and the tunnel server. This configuration is for L2TP tunnels.

Local Peer Configuration

```
l2tp-class l2tpclass1
!
pseudowire-class pwclass1
 encapsulation l2tpv2
 protocol l2tpv2 l2tpclass1
 ip local interface ethernet0/0
!
interface virtual-ppp 1
 ip unnumbered loopback1
 ppp authentication chap
 ppp chap hostname peer1
 pseudowire 172.24.13.196 10 pw-class pwclass1
!
ip route 10.10.10.0 255.255.255.0 virtual-PPP 1
```

Tunnel Server Configuration

```
vpdn-group l2tpgroup1
 accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname peer1
!
interface virtual-template 1
 ip unnumbered loopback 1
 ppp authentication chap
 ppp chap hostname peer2
```

Example Configuring L2TPv3 Client-Initiated Tunneling

The following example configures L2TP client-initiated tunneling on the local peer and tunnel server. This configuration is for L2TPv3 tunnels.

Local Peer Configuration

```
l2tp-class l2tpclass1
```

```
!
pseudowire-class pwclass1
 encapsulation l2tpv3
 protocol l2tpv3 l2tpclass1
 ip local interface ethernet0/0
!
interface virtual-ppp 1
 ip unnumbered loopback1
 ppp authentication chap
 ppp chap hostname peer1
 pseudowire 172.24.13.196 15 pw-class pwclass1
!
ip route 10.10.10.0 255.255.255.0 virtual-PPP 1
```

Tunnel Server Configuration

```
l2tp-class l2tpclass2
!
pseudowire-class pwclass2
 encapsulation l2tpv3
 protocol l2tpv3 l2tpclass2
 ip local interface ethernet0/1
!
interface virtual-ppp 2
 ip unnumbered loopback 1
 ppp authentication chap
 ppp chap hostname peer2
 pseudowire 172.16.32.24 15 pw-class pwclass2
!
ip route 10.20.20.0 255.255.255.0 virtual-PPP 1
```

Example Verifying an L2TP Control Channel

The following output displays detailed information the L2TP control channels that are set up to other L2TP-enabled devices for all L2TP sessions on the router:

```
Router# show l2tun session all
Session Information Total tunnels 0 sessions 1
Session id 111 is up, tunnel id 0
Call serial number is 0
Remote tunnel name is
  Internet address is 2.0.0.1
  Session is manually signalled
  Session state is established, time since change 00:06:05
    0 Packets sent, 0 received
    0 Bytes sent, 0 received
    Receive packets dropped:
      out-of-order:             0
      total:                    0
    Send packets dropped:
      exceeded session MTU:     0
      total:                    0
  Session vcid is 123
Session Layer 2 circuit, type is ATM VPC CELL, name is ATM3/0/0:1000007
Circuit state is UP
    Remote session id is 222, remote tunnel id 0
  DF bit off, ToS reflect disabled, ToS value 0, TTL value 255
  Session cookie information:
    local cookie, size 8 bytes, value 00 00 00 00 00 00 00 64
    remote cookie, size 8 bytes, value 00 00 00 00 00 00 00 C8
```

```
  SSS switching enabled
Sequencing is off
```

Example Configuring Client-Initiated VPDN Tunneling Using PPTP

The following example shows the configuration of a tunnel server for client-initiated VPDN tunneling with the PPTP protocol using an ISA card to perform stateless MPPE encryption:

```
vpdn-group pptp1
accept-dialin
  protocol pptp
  virtual-template 1
 local name cisco_pns
!
interface virtual-template 1
 ip unnumbered FastEthernet 0/0
 peer default ip address pool mypool
 encapsulation ppp
 ppp authentication ms-chap
 ppp encrypt mppe auto
!
controller ISA 5/0
 encryption mppe
```

Where to Go Next

You can perform any of the relevant optional tasks in the Configuring Additional VPDN Features and in the VPDN Tunnel Management modules.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
VPDN commands	*Cisco IOS VPDN Command Reference*
VPDN technology overview	VPDN Technology Overview module
Information about virtual templates	Configuring Virtual Template Interfaces module
Dial Technologies commands	*Cisco IOS Dial Technologies Command Reference*
Technical support documentation for L2TP	*Layer 2 Tunnel Protocol (L2TP)*
Technical support documentation for PPTP	*Point to Point Tunneling Protocol (PPTP)*
Technical support documentation for VPDNs	*Virtual Private Dial-Up Network (VPDN)*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
• CISCO-VPDN-MGMT-MIB • CISCO-VPDN-MGMT-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2637	Point-to-Point Tunneling Protocol (PPTP)
RFC 2661	*Layer Two Tunneling Protocol L2TP*
RFC 3931	*Layer Two Tunneling Protocol - Version 3 (L2TPv3)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Client-Initiated VPDN Tunneling

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

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

Table 1: Feature Information for Client-Initiated VPDN Tunneling

Feature Name	Software Releases	Feature Configuration Information
L2TP Client-Initiated Tunneling	12.3(2)T	This feature introduces the ability to establish client-initiated L2TP tunnels. The client can initiate an L2TP or L2TPv3 tunnel to the tunnel server without the intermediate NAS participating in tunnel negotiation or establishment. The following commands were introduced or modified by this feature: **authentication** (L2TP), **encapsulation** (L2TP), **hello**, **hidden**, **hostname** (L2TP), **interface virtual-ppp**, **ip dfbit set**, **ip local interface**, **ip pmtu**, **ip protocol**, **ip tos** (L2TP), **ip ttl**, **l2tp-class**, **password** (L2TP), **protocol** (L2TP), **pseudowire**, **pseudowire-class**, **receive-window**, **retransmit**, **sequencing**, **timeout setup**.