Configuring L2TPv3 Over UDP/IP

Layer 2 Tunneling Protocol (L2TPv3), is a tunneling protocol that enables tunneling of Layer 2 packets over IP core networks.

L2TPv3 tunnel is a control connection between the end points. One L2TPv3 tunnel can have multiple data connections, and each data connection is termed as an L2TPv3 session. The control connection is used to establish, maintain, and release sessions. Each session is identified by a unique session ID.

To provide the tunneling service to Ethernet traffic, L2TPv3 feature employs:

- L2TPv3
- Pseudowire (PW) technology

Prerequisites

These are the prerequisites for configuring L2TPv3:

- IP routing must be enabled before configuring L2TP-class

  This command enables IP routing:

  ip routing

- IP CEF must be enabled

  This command enables IP CEF:

  ip cef

- Subinterfaces for Vlans must be created

  These commands create subinterfaces for VLANs:

  interface Dot11Radio ***interface number.sub-interface number***

  encapsulation dot1Q ***vlan id***

  bridge-group ***bridge id***

  interface GigabitEthernet0.***sub-interface number***

  encapsulation dot1Q ***vlan id***

  bridge-group ***bridge id***

Note The bridge id on interfaces with same vlan id must be the same.

The following are not supported:

- Tunnel establishment using IPv6 address
- SNMP and GUI configuration
- Multiple tunnels to same LNS (L2TP Network Server)
- Configuring xconnect on physical interfaces like Gig and Dot11
- Prol2tp versions older than 1.6.1 when sequencing or cookies are enabled.
- Xconnect allows only IPv4 address. FQDN is not supported.
- Only dynamic cookie assignment is used.

Configuring L2TP Class

Configuring the L2TP creates a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes. These parameters can be configured:

- Authentication
- L2TPv3 hello interval
- Hostname
- Cookie length
- Enabling digest
- Retransmit and retries for the L2TPv3 control packets
- Timeout
- Receive-window size
- Hello interval

Beginning in privileged EXEC mode, follow these steps to configureL2TP Class

	Command	Purpose
Step 1	**digest hash** *[MD5, SHA]*	enable message digest.
Step 2	**receive-window** *size*	Receive window size of control connection.
Step 3	**hello** *interval*	Configure the interval between two hello messages.
Step 4	**cookie size** *cookie size*	Configure the cookie size. The values are 4 and 8.
Step 5	**digest secret** *secret*	Configure the secret for authentication.
Step 6	**retransmit retries** *retries*	Configure the number of times a control message is sent if no response is received.
Step 7	**retransmit timeout min** *minimum timeout*	Configure the minimum timeout beween retries.
Step 8	**retransmit timeout max** *maximum timeout*	Configure the maximum timeout between retries.

Note Multiple l2tp classes can be configured.

Examples

```
ap1# configure terminal
ap1(config)# l2tp-class myl2tpclass
ap1(config-l2tp-class)# hostname myhost1
ap1(config-l2tp-class)# hello 15
ap1(config-l2tp-class)# cookie size 4
ap1(config-l2tp-class)# digest secret cisco
ap1(config-l2tp-class)# retransmit retries 6
ap1(config-l2tp-class)# retransmit timeout 7
ap1(config-l2tp-class)# retransmit timeout max 5
ap1(config-l2tp-class)# retransmit timeout min 1
ap1(config-l2tp-class)# end
```

Configuring Pseudowire Class

Configuring the pseudowire class defines a layer 2 pseudowire class. These pseudowire parameters can be configured under pseudowire class:

- encapsulation method
- l2tp-class
- local interface
- sequencing
- IP related parameters like dfbit, tos and ttl

Beginning in privileged EXEC mode, follow these steps to configure Pseudowire Class

	Command	Purpose
Step 1	**pseudowire-class** *pseudowire class name*	Specifies the pseudowire class name.
Step 2	**encapsulation l2tpv3**	Enables the L2TPv3
Step 3	**protocol l2tpv3ietf** *l2tp class name*	Enables the standard L2TPv3 and attaches the L2TP class.
Step 4	**ip protocol udp**	Enables L2TPv3 over UDP.
Step 5	**ip local interface** *interface name*	Uses the interface address as the source address.

Examples

```
ap1# configure terminal
ap1(config)# pseudowire-class mypwclass
ap1(config-pw-class)# encapsulation l2tpv3
ap1(config-pw-class)# protocol l2tpv3ietf myl2tpclass
ap1(config-pw-class)# ip protocol udp
ap1(config-pw-class)# ip local interface BVI1
ap1(config-pw-class)# end
```

Relationship between L2TP Class and Pseudowire Class

Multiple pseudowire classes can be configured. A pseudowire class can configured with any one of the available L2TP Classes. Xconnect can be configured with any one of the configured pseudowire classes.

The following points should be kept in mind:

- A pseudowire class can have only one L2TP Class attached to it.
- An L2TP Class can be attached to multiple pseudowire-classes.
- An xconnect command has a pseudowire-class attached to it, so for one xconnect command only one pseudowire and one L2TP Class is sufficient.
- An L2TP Class not attached to a pseudowire-class and a pseudowire not attached to a xconnect command have no effect on working of an AP.
- L2TP Class attached with a Pseudowire Class cannot be modified. To modify, remove the xconnect from interface which is using this Pseudowire Class.

Configuring the Tunnel interface

This is a new interface for single tunnel support. You can configure xconnect here for all L2TPv3 traffic.

Beginning in privileged EXEC mode, follow these steps to configure the tunnel interface:

	Command	Purpose
Step 1	**interface VDT** *index*	Specifies the VDT interface.
Step 2	**no ip address**	Disables the IP addresses
Step 3	**xconnect** *LNS ip* | *vc-id* | **pw-class** *pseudowire class name*	Configures the LNS IP and attaches the Pseudowire Class.

The vc id is a number which is locally significant. Every xconnect command must be configured with a unique vc id. Traffic for ssids that have **xconnect** *VDT index* configured, get tunneled through a VDT interface with same index.

Examples

```
ap1# configure terminal
ap1(config)# interface VDT0
ap1(config-if)# xconnect 100.100.10.2 10 pw-class mypwclass
ap1(config-if)# end
```

Configure Tunnel management Interface

This is a new interface for secondary tunnel support.

Beginning in privileged EXEC mode, follow these steps to configure the tunnel management interface:

	Command	Purpose
Step 1	**interface VDT-Mgmt** index	Specifies the VDT management interface.
Step 2	**no ip dhcp client request router**	Disables the default route from dhcp.
Step 3	**ip address** *dhcp* | *ip netmask*	Specifies the dhcp IP or static IP.
Step 4	**vdt-mgmt vlan 10**	Configures the VLAN id.

This interface allows access to an AP through the tunnel. This interface is associated with a VDT interface with same index. Traffic from this interface is tunneled though a tunnel established with VDT interface with same index.

Note There will be two default routes leading to a communication failure if the default route from dhcp is not disabled using the **no ip dhcp client request router** command.

Examples

```
ap1# configure terminal
ap1(config)# interface VDT-Mgmt0
ap1(config-subif)# no ip dhcp client request router
ap1(config-subif)# ip address dhcp
ap1(config-subif)# vdt-mgmt vlan 10
ap1(config)# end
```

Mapping SSID to the Tunnel/Xconnect

Mapping the tunnel to the WLAN is done by adding Xconnect under the ssid configuration.

Beginning in privileged EXEC mode, follow these steps to map the tunnel to the VLAN:

	Command	Purpose
Step 1	**dot11 ssid** *ssid*	Specifies the ssid.
Step 2	**vlan** *vlan id*	Specifies the VLAN id.
Step 3	**xconnect** *index of VDT interface*	Enables L2TPv3 for the ssid.
Step 4	**authentication open**	Specifies the type of authentication.

Examples

```
ap1# configure terminal
ap1(config)# dot11 ssid myssid
ap1(config-ssid)# vlan 10
ap1(config-ssid)# authentication open
ap1(config-ssid)# xconnect 0
ap1(config-ssid)# end
```

Configuring TCP mss adjust

To configure TCP mss adjust for tunnel clients use the **dot11 l2tp tcp mss** *tcp mss value command* in the configuration mode.

dot11 l2tp tcp mss *tcp mss value*

Examples

```
ap# configure terminal
ap(config)# dot11 l2tp tcp mss 1360
ap1(config)# end
```

Configuring UDP checksum

To configure UDP checksum ignore for fragmented L2TPv3oUDP Data Packets use the **dot11 l2tpoUdp udp checksum zero** in the configuration mode.

dot11 l2tpoUdp udp checksum zero

Note This command is used when the prol2tp server version is older than 1.6.1 are used.

Examples

```
ap# configure terminal
ap(config)# dot11 l2tpoUdp udp checksum zero
ap(config)# end
```