Technical Guide

Allied Telesis™

L2TPv2

Feature Overview and Configuration Guide

This document describes the L2TPv2 capabilities supported by the AR-Series firewalls. The AR-Series firewalls support operation in two types of L2TP tunnel:

- as a LAC for LAC-LNS tunnels
- peer-to-peer tunnels

The first section of the document provides an overview of L2TP.

The next section describes the LAC-LNS tunnels, and how the LAC tunnels PPP sessions from the client PPPoE connections. There are two main sections to this description, followed by configuration information:

- an overview of LAC-LNS L2TPv2 tunnels including their connection processes ("L2TPv2 Tunnels" on page 4).
- an explanation of how the LAC end of an L2TP tunnel is used for PPPoE client connections and tunneling their PPP sessions via the L2TP LAC to remote L2TP LNS devices ("PPPoE Access Concentrator (AC)" on page 13).
- an explanation of how the LAC end of the L2TP tunnel employs PPPoE AC functionality to transition multiple PPPoE connections onto an L2TPv2 tunnel ("PPPoE Access Concentrator (AC)" on page 13).
- configuration examples showing how to configure an AlliedWare Plus device to terminate multiple incoming PPPoE client connections and tunnel their PPP sessions via the L2TP LAC to one or more remote L2TP LNS devices. They illustrate how to use DNS or RADIUS lookups or static configuration to determine the destination's LNS. See "L2TPv2 LAC-LNS Tunnel Configuration" on page 15.

The final section describes support for Managed L2TPv2 peer-to-peer VPNs, including:

- an overview explanation of how the L2TPv2 protocol can be used to establish a peer-to-peer Layer 3 VPN over an intermediate IP network ("Managed L2TPv2 Peer-to-Peer Tunnels" on page 25).
- a series of configuration examples, including interoperation with AlliedWare and also L2TP with IPsec protection ("Managed L2TPv2 Peer-to-Peer Tunnel Configuration" on page 27).

AlliedWare Plus™
OPERATING SYSTEM

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support PPP tunneling via L2TPv2 (LAC), running version **5.4.6** or later.

To see whether your product supports PPP tunneling via L2TPv2 (LAC), see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

Managed L2TPv2 peer-to-peer tunnels are available from 5.4.6-1.1 onwards.

Feature support may change in later software versions. For the latest information, see the above documents.

Related documents

The following documents give more information about related features on AlliedWare Plus products:

- the Point-to-Point Protocol (PPP) Feature Overview and Configuration Guide
- The RADIUS Feature Overview and Configuration Guide
- the DNS section in the Internet Protocol (IP) Addressing and Protocols Feature Overview and Configuration Guide

These documents are available from the links above or on our website at alliedtelesis.com

L2TPv2 Tunnels

This sections describes features of L2TPv2 tunnels. Subsequent sections describe the L2TPv2 features supported by AR-Series firewalls.

Overview

The L2TPv2 components implemented on AlliedWare Plus devices supporting L2TPv2 tunnels are based on the functionality described in RFC 2661.

L2TPv2 tunnels Layer 2 connections through a separate or intermediate Layer 3 network such as the Internet.

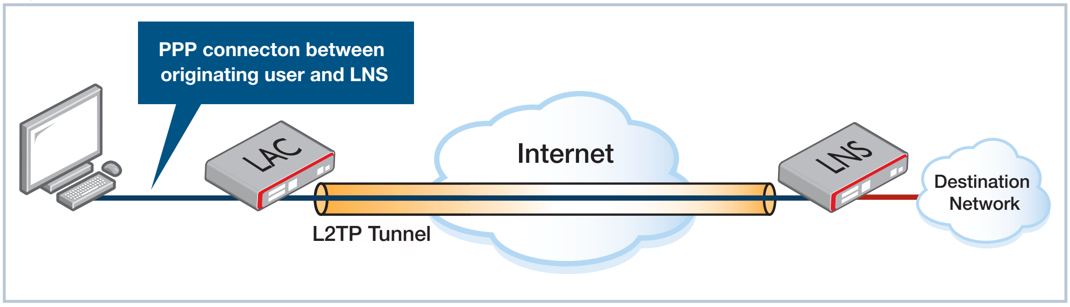
The L2TP protocol can be used to establish two quite separate types of tunnels:

- LAC-LNS tunnels
- peer-to-peer tunnels

LAC-LNS tunnels

L2TPv2 can create tunnels between a local endpoint (an L2TP Access Concentrator or LAC) and a remote endpoint (an L2TP Network Server or LNS). The LAC typically acts as a client and initiates the tunnels, while the LNS typically acts as a server that listens for incoming tunnel requests. Network traffic is bi-directional between the LAC and the LNS once an L2TP tunnel is established through the intermediate network. See Figure 1 below.

Figure 1: L2TP to tunnel PPP



The LAC creates an L2TPv2 tunnel to the defined LNS. The connection request from the LAC may include the information required to allow the LNS to authenticate the user at the originating end of the Layer 2 connection and accept or decline the connection. Once the L2TPv2 tunnel is established, an L2TP session is created over the tunnel. Encapsulated PPP frames associated with the session can then pass through the tunnel. The LNS accepts the frames, strips off the L2TP encapsulation and processes them as normal incoming PPP frames. These PPP frames are processed as if they had come directly from the link layer. For an L2TP and PPP encapsulation diagram for data over an IP network see Figure 2 on page 7.

Peer-to-peer tunnels

Managed peer-to-peer tunnels allow for the establishment of a single negotiated peer-to-peer Layer 3 VPN over an intermediate IP network.

The L2TPv2 VPN encapsulates PPP frames, allowing a PPP link to be established directly between the two peers.

L2TPv2 managed peer-to-peer tunnels are stateful (as are LAC-LNS tunnels) and hence are aware of the status of the remote node and will only pass traffic if the remote node is reachable. The encapsulated PPP layer also offers several advantages, such as PAP or CHAP-based user authentication, and the ability to assign an IP address to the remote node PPP interface dynamically.

L2TPv2 terminology

This section describes some key L2TPv2 terms.

- **L2TPv2 (Layer 2 Tunneling Protocol)**

  L2TPv2 enables encapsulated Layer 2 frames to be carried across the network. For L2TPv2 these L2 frames are Point-to-Point Protocol (PPP) Layer 2 frames. L2TPv2 enables Point-to-Point (PPP) frames to be totally encapsulated within network packets, so that they can then be tunnelled through a Layer 3 network such as the Internet. PPP defines an encapsulation mechanism for transporting multiprotocol packets, such as IP packets, across point-to-point links. L2TPv2 extends the PPP model by tunneling the point-to-point link across an intermediate Layer 3 network.

- **LAC (L2TP Access Concentrator)**

  An LAC resides at one end of an L2TPv2 tunnel. The LAC sits between an LNS and a client and forwards PPP packets to and from each. Packets sent from the LAC to the LNS are encapsulated and sent into the L2TP tunnel. The packets sent from the LNS arrive via the tunnel, are decapsulated, and sent to the client. The LAC and LNS have no awareness of what data is contained within the PPP packets.

- **LNS (L2TP Network Server)**

  An LNS resides at one end of an L2TPv2 tunnel and acts as a peer to the LAC. An LNS is an L2TPv2 server that terminates the incoming tunnel from the L2TP LAC. An LNS is the logical termination point of the PPP session that is being tunneled from the client by the LAC.

- **L2TPv2 Tunnel**

  A tunnel is a logical connection between the LAC and the LNS, or between two peers, that can carry PPP sessions. In the case of an LAC-LNS tunnels, a tunnel can carry multiple sessions. In the case of a peer-to-peer tunnel, a tunnel can carry only one PPP session. The tunnel consists of a control connection and zero or more sessions, each of which carries an encapsulated PPP connection. With AlliedWare Plus L2TPv2, each tunnel has two tunnel identifiers, one for the local end and one for the peer. Outgoing frames have the destination tunnel identifier.

- **L2TPv2 Session**

  An L2TPv2 session is created between the LAC and LNS when an end-to-end PPP connection is to be established between a client and the LNS. Similarly, an L2TP session is created between two peers to carry the PPP connection that joins these peers. There is a one-to-one relationship between established L2TPv2 sessions and their associated L2TPv2 calls, where a single tunnelled PPP session is referred to as an L2TPv2 call. An L2TPv2 session must be created before the PPP session can be established since L2TPv2 is connection-oriented. L2TPv2 is a connection oriented protocol in that both the LAC and its associated LNS each manage and maintain the state of their L2TPv2 connection.

L2TPv2 tunnel messages

L2TP has two types of messages: control messages and data messages. Control messages are used for tunnel establishment, tunnel maintenance and session management. Data messages are used to encapsulate PPP frames.

- Control messages constitute communication between the LAC and LNS. The control messages are 'in-band' in that they use the same packet transport that is used for data packets. The data within the control messages is carried as a series of AVPs.
- An AVP (Attribute Value Pair) is a variable length concatenation of an attribute ID and an associated value for the attribute. Multiple AVPs make up control messages that are used in the establishment, maintenance and teardown of L2TP tunnels.
- Some control messages contain no AVPs. These 'empty' messages are called ZLB (Zero Length Body) messages, so these are L2TP control packet with only an L2TP header. ZLB messages are used for acknowledging other control messages.

Control message types

There are two sets of L2TPv2 control messages:

- The messages that control L2TP tunnels (Table 1)
- The messages that control L2TP calls (sessions) within a tunnel (Table 2)

Table 1: L2TP tunnel (control channel) connection management messages that are used to establish, clear, and maintain L2TP tunnels

MESSAGE TYPE	MESSAGE NAME
1 (SCCRQ)	Start-Control-Connection-Request
2 (SCCRP)	Start-Control-Connection-Reply
3 (SCCRN)	Start-Control-Connection-Connected
4 (StopCCN)	Stop-Control-Connection-Notification.

Table 2: L2TP session or call management messages that are used to establish, clear, and maintain L2TP calls (sessions) within the tunnels

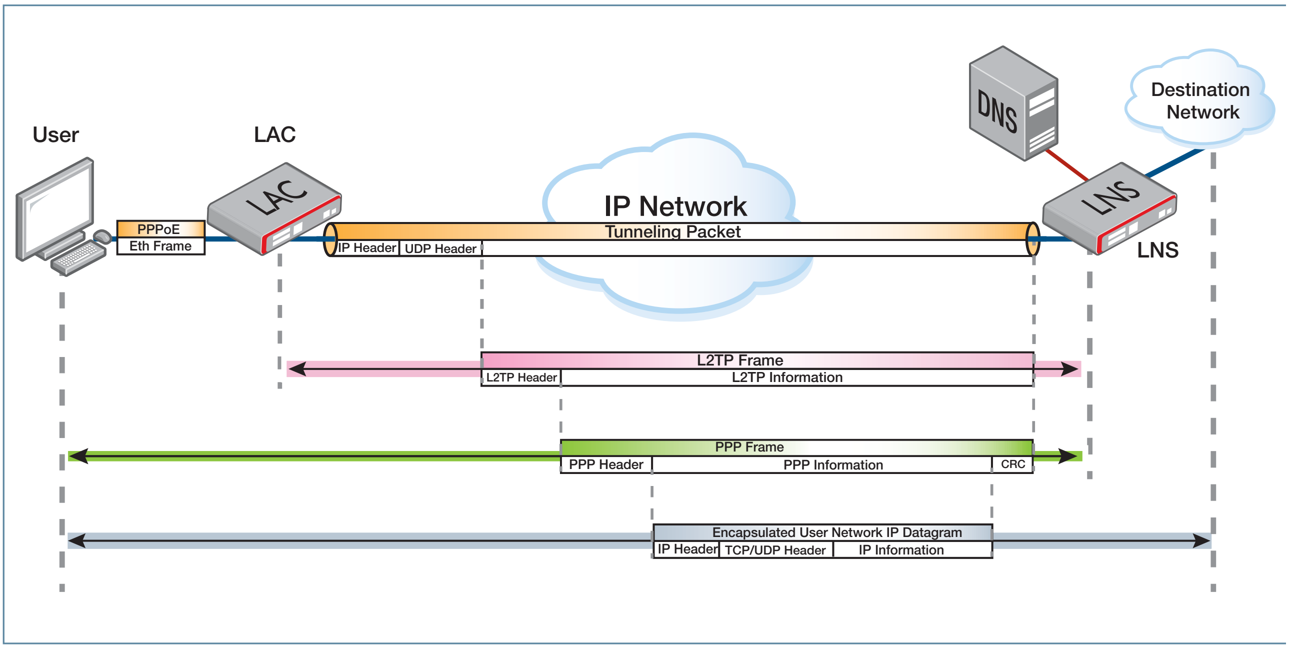
MESSAGE TYPE	MESSAGE NAME
10 (ICRQ)	Incoming-Call-Request
11 (ICRP)	Incoming-Call-Reply
12 (ICCN)	Incoming-Call-Connected
14 (CDN)	Call-Disconnect-Notify.

For the messaging sequence when establishing an L2TP connection, see Figure 3 on page 9.

LAC-LNS Tunnels

L2TPv2 encapsulation for LAC-LNS tunnels

Figure 2: L2TPv2 encapsulation: PPP frame encapsulated within L2TP frame within tunneling packet



Process of establishing an LAC-LNS connection

The LAC and LNS go through a negotiation process to establish the tunnel. The negotiation process involves each peer sending a number of messages that contain a series of settings, communicated in AVPs.

The sequence for a typical successful L2TP connection between the LAC and the LNS is listed below. See "L2TPv2 tunnel messages" on page 6 for a description of the message types.

1. The LAC sends an SCCRQ (Start-Control-Connection-Request) message to the LNS. AVPs (Attribute Value Pairs) are included in this message.The AVPs contain settings that the LAC would like to have applied to the tunnel.
2. The LNS responds to the LAC with an SCCRP (Start-Control-Connection-Reply) message. The response to the LAC challenge and AVPs are included with this message.The AVPs indicate the settings it is requesting.

   If either the LAC or the LNS has an issue with the settings supplied by the other, then it will tear down the tunnel with a StopCCN (Stop-Control-Connection) message.
3. The LAC sends an SCCCN (Start-Control-Connection-Connected) message to the LNS.

   A shared secret may optionally be configured for extra security. If a shared secret is configured on both peers, the initiator will include a CHAP challenge in the SCCRQ packet. The recipient will then reply to the CHAP response as well as send its own CHAP challenge in the SCCRP packet. The initiator will send a CHAP response in the SCCCN packet. If one peer is not configured with a shared secret or if the peers are configured with different shared secrets then the tunnel will be torn down with a StopCCN packet.
4. The LNS responds to the LAC with a ZLB ACK (Zero Length Body Acknowledgement) message. The ZLB ACK message may be in another message. The L2TP tunnel is now up.
5. The setting up of the call within the tunnel now begins. The LAC sends an ICRQ (Incoming-Call-Request) message to the LNS. The ICRQ contains a number of requested settings for the recipient to use for the session, as well as a session ID, which is a locally significant value used by the LNS to identify the session.
6. The LNS responds to the LAC with an ICRP (Incoming-Call-Reply) message. The ICRP contains settings for the LAC to use as well as a session ID for the LAC to use as an identifier for the session.

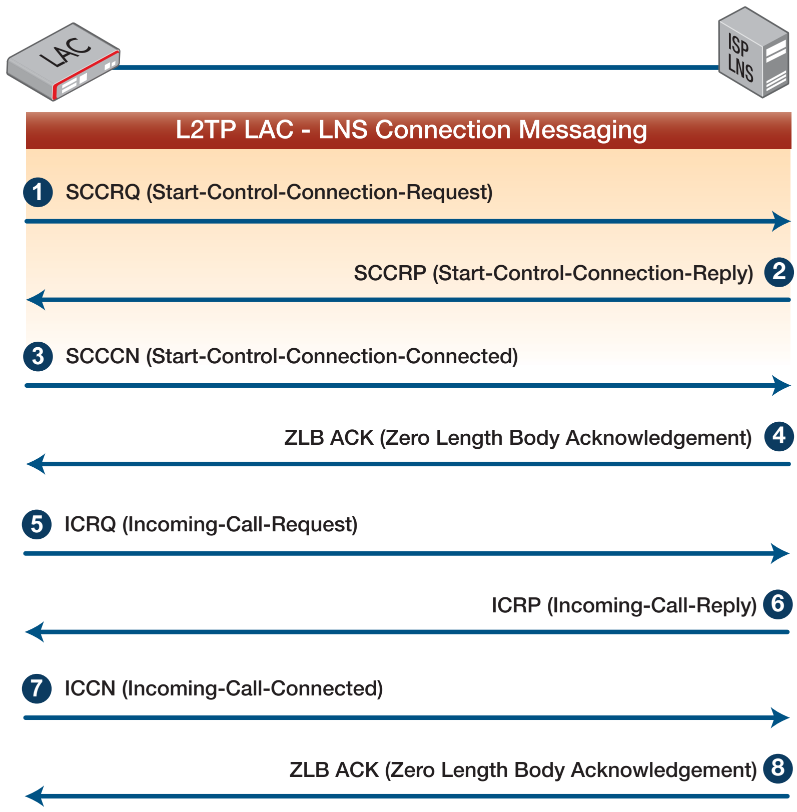
   If either the LAC or the LNS has an issue with the settings supplied by the remote node then it will tear down the tunnel with a CDN (Call-Disconnect-Notify) message.

7. The LAC send an ICCN (Incoming-Call-Connected) message to the LNS.

8. The LNS responds to the LAC with a ZLB ACK (Zero Length Body Acknowledgement) message. The ZLB ACK message may be in another message. The L2TP session is now up.

9. PPP negotiation can now begin after both the L2TP tunnel and the L2TP session are up. Figure 5 on page 11 shows the process for connecting a PPP client. Figure 3 below corresponds to Steps 4-5 in that diagram.

Figure 3: L2TP LAC - LNS Connection Messaging Diagram


LAC
ISP LNS
L2TP LAC - LNS Connection Messaging
1 SCCRQ (Start-Control-Connection-Request)
SCCRP (Start-Control-Connection-Reply) 2
3 SCCCN (Start-Control-Connection-Connected)
ZLB ACK (Zero Length Body Acknowledgement) 4
5 ICRQ (Incoming-Call-Request)
ICRP (Incoming-Call-Reply) 6
7 ICCN (Incoming-Call-Connected)
ZLB ACK (Zero Length Body Acknowledgement) 8

Whilst the tunnel is established, the LAC will send L2TP keepalive messages at regular intervals, to which the LNS is expected to respond. If the LAC does not receive a reply within a set amount of time, it will consider the tunnel down and will automatically attempt to reestablish it.

Operation of the L2TP LAC

As previously discussed, the LAC can transport one or more PPP client sessions inside an L2TP tunnel to an LNS device. In fact, a given LAC can connect to multiple LNSs. So different PPP connections being tunneled by the LAC can be tunneled to different LNSs.

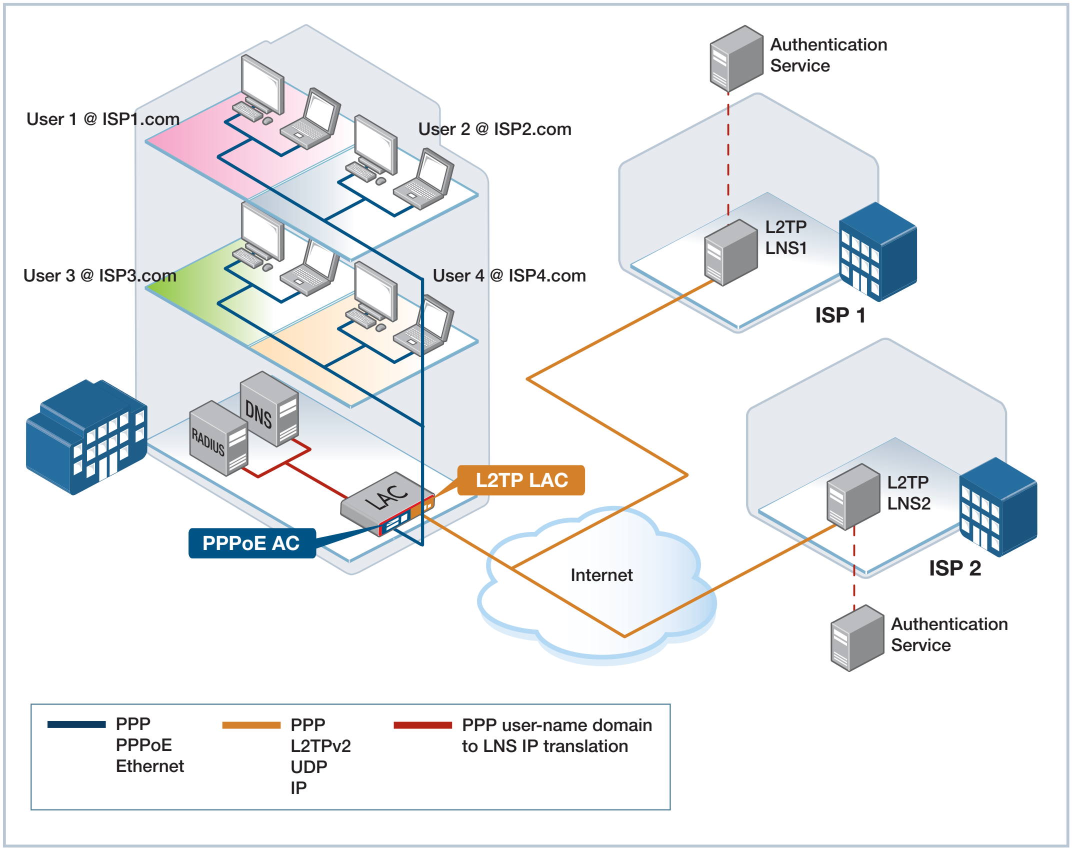
The LAC determines the target IP address of the LNS for each PPP's tunnel via one of several methods:

- By statically configuring the IP address of the LNS.
- By using the domain information extracted from each PPP client session and performing DNS lookup.
- By using the domain information extracted from each PPP client session and performing RADIUS lookup.

When a client PPP session arrives at the LNS device, the LNS extracts it from the L2TP tunnel and fully terminates and authenticates it. The ISP managing the LNS typically allocates an Internet IP address to the client PPP interface from its own database.

In the diagram below, multiple clients who wish to connect to different ISPs create PPPoE connections to the same LAC. By extracting the domain credentials from the users' sessions, the LAC determines the LNS to which to tunnel each user's PPP connection.

Figure 4: Example network using L2TPv2 to tunnel PPP

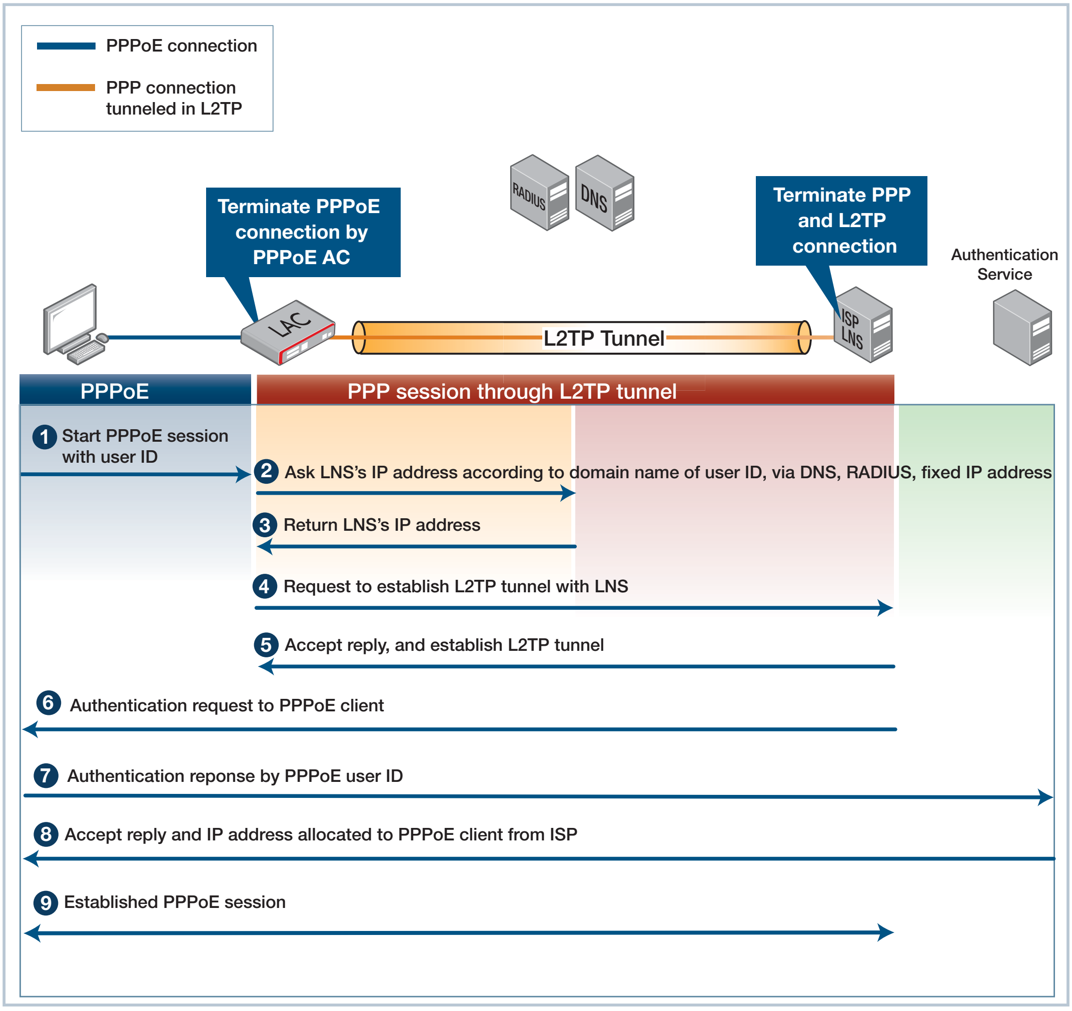


Process of establishing a PPP connection through a tunnel

The purpose of L2TP tunneling is to form an end-to-end PPP connection that is terminated at one end by a client and at the other end by an LNS. A common scenario is for home-user clients connecting via PPP to an LNS at an ISP's premises. The AR-series Firewall can provide the PPPoE Access Concentrator and the L2TP LAC that work together to take the client's PPPoE connection, and tunnel it through to an LNS.

The sequence of events involved in establishing the connection from the home user through to the LNS are illustrated in Figure 5 below.

Figure 5: Establishing an L2TPv2 tunnel for PPP


PPPoE connection
PPP connection tunneled in L2TP
Terminate PPPoE connection by PPPoE AC
RADIUS
DNS
Terminate PPP and L2TP connection
Authentication Service
LAC
L2TP Tunnel
ISP LNS
PPPoE
PPP session through L2TP tunnel
1 Start PPPoE session with user ID
2 Ask LNS's IP address according to domain name of user ID, via DNS, RADIUS, fixed IP address
3 Return LNS's IP address
4 Request to establish L2TP tunnel with LNS
5 Accept reply, and establish L2TP tunnel
6 Authentication request to PPPoE client
7 Authentication reponse by PPPoE user ID
8 Accept reply and IP address allocated to PPPoE client from ISP
9 Established PPPoE session

Note that the PPP connection is established between the client and the LNS. Steps 4-5 in this diagram correspond to the connection sequence shown in Figure 3 on page 9.

The process of establishing such a connection using L2TP tunneling of PPPoE is transparent to the PPP client, and is as follows:

1. The client initiates a PPP connection over PPPoE to the LAC—the AR-series firewall that is acting as the PPPoE Access Concentrator and L2TP LAC.
2. The LAC requests the IP address of the ISP’s L2TP Network Server (LNS) from static configuration or by RADIUS or DNS lookup. This is the address of the LNS that will terminate the PPP session.
3. The DNS or RADIUS server returns the IP address of the LNS.
4. Once the LAC gets the IP address to the LNS, it acts as an L2TP LAC, and sends a request to the LNS to establish an L2TP tunnel. The authentication for the tunnel will depend on the setting of the ‘shared-secret’. When a shared-secret is given, authentication mode is set to challenge; when shared-secret is not given, authentication mode is set to none.

   If the LAC issues an EAP or CHAP challenge, the name field for the challenge will be populated with a dummy name (see more explanation of this in "Details of the implementation of LAC-LNS tunnels on the AR-Series firewalls" on page 13).

   The L2TPv2 Vendor name AVP will be populated with text identifying the vendor (this is described some more in "Details of the implementation of LAC-LNS tunnels on the AR-Series firewalls" on page 13).

   If the LNS address was determined by RADIUS or DNS query, the LAC packages up the PPP information it learned from PPPoE initiation packets via the partial termination. It passes this information to the LNS as a 'proxy auth’ pair via the ICCN message.
5. The LNS replies and establishes the L2TP tunnel to the LAC.
6. In establishing the L2TPv2 tunnel (after DNS or RADIUS lookup) the L2TPv2 ICCN message will be populated with PPP Proxy Auth AVPs [Attribute-Value Pairs] filled with information from the already partially terminated client PPP sessions (for instance challenge and response for CHAP). The LNS should be configured to use those AVPs to take over the PPP link establishment itself and in doing so authenticate (by which ever means it chooses) the PPP client. This will be on by default and will support CHAP, PAP and EAP.
7. The LNS requests PPP authentication from it’s authentication service.
8. The authentication service authenticates the PPP client and allocates an IP address to the PPP client.
9. The PPP client, the LAC and the LNS have now formed an end-to-end PPP connection. The client packages data into PPP frames that it sends over PPPoE to the LAC, which then strips the PPPoE layer and forwards the contained PPP packets through the L2TP tunnel to the ISP’s LNS.

If the PPP negotiation fails, then they tear down the underlying L2TP session with a CDN packet.

Details of the implementation of LAC-LNS tunnels on the AR-Series firewalls

The functionality of L2TPv2 LAC-LNS tunnels that is supported on your device is implemented as defined in RFC 2661, ‘Layer Two Tunneling Protocol (L2TPv2)’. Your device can act as an L2TPv2 LAC, but not as an LNS.

- By default, connections are established via the control channel using UDP destination port 1701. Note that L2TPv3 unmanaged pseudowires also use UDP port 1701 by default. If both L2TPv2 tunnels and L2TPv3 pseudowires are to be used simultaneously, then the UDP port of the L2TPv3 pseudowires must be changed. Use the command:

  ```
  awplus(config)# l2tp unmanaged port <1-65535>
  ```

  We recommend that the port used be outside the range of well-known ports. We recommend against using a well known port that could be used by a common service that could be enabled in the future.
- Up to 256 concurrent PPP sessions are supported across L2TPv2 tunnels.
- Up to 6 concurrent L2TPv2 LAC to LNS tunnels are supported.
- If the method for determining the LNS IP address uses either DNS or RADIUS lookups, the PPP user-name must be in the format ‘user@domain.com’.
- If the LAC issues an EAP or CHAP challenge, the name field for the challenge will be populated with the text 'Allied Telesis'.
- The L2TPv2 Vendor name AVP will be populated with the text ‘Allied Telesis Inc’.
- If a shared secret is configured, AlliedWare Plus uses a 16 byte MD5 hash for the CHAP challenge, which is compatible with most other vendors’ implementations. Vendor implementations that use a 32 byte hash are not compatible.

PPPoE Access Concentrator (AC)

As described above, the AR-Series firewall acts as a PPPoE access concentrator, to enable clients to establish a PPPoE session that is ultimately tunneled through to the LNS.

It is common for clients to access the Internet via a PPPoE connection. You can think of a PPPoE Access Concentrator (AC) as a possible termination point of one or more PPPoE client sessions.

The AR-Series firewall PPPoE Access Concentrator (AC) can terminate multiple incoming PPPoE client connections and tunnel their PPP sessions via the L2TP LAC to one or more remote L2TP LNS devices.

The user typically configures each PPPoE client with both:

- a PPPoE service-name to connect to a PPPoE Access Concentrator
- a PPP user ID in the format of username@domain, e.g. john@ISP1.com, to allow authentication.

The PPPoE AC needs to be configured with a matching service name. The PPPoE AC on the AR-series Firewall can temporarily terminate and extract the domain portion information from each client PPP session. It uses the domain information to determine which ISP each client wishes to connect to via an L2TP tunnel.

The AR-Series firewall only supports operating as a PPPoE AC for the purposes of tunneling PPP client sessions to a remote ISP via L2TP; it cannot operate as a fully functional PPPoE access concentrator.

DNS and RADIUS

The LAC can be configured to use DNS or RADIUS lookup to get the IP address of the LNS.

DNS

If DNS lookup is configured, the LAC partially terminates the PPP session from the client to retrieve the user-name of the session. For a PPP user-name of, for instance, user@domain.com, the LAC extracts the 'domain.com' portion and prefixes it with a configurable string, such as ‘lns’. This string can be can be configured by the command **l2tp peer-address dns-lookup prefix <*string*>**. It sends this in a DNS request to the DNS server, which sends back the IP address of the LNS.

RADIUS

If RADIUS lookup is configured, the LAC again partially terminates the PPP session from the client to retrieve the user-name of the session. For a PPP user-name of, for instance, user@domain.com, the LAC extracts the 'domain.com' portion. It then sends an Access-Request packet to the RADIUS server with parameters:
User-Name=domain.com,User-Password=’password’.

The RADIUS server responds by sending an Access-Accept packet back to the LAC with the IPv4 address of the LNS contained in Framed-IP-Address format (e.g. 192.168.11.2). Note that the RADIUS server is doing no authentication in this process.

L2TPv2 LAC-LNS Tunnel Configuration

This section describes what needs to be configured for L2TPv2 tunneling of PPPoE, followed by configuration procedures for:

- "Example 1: Tunneling PPPoE connection with a static L2TP LNS address" on page 16
- "Example 2: Tunneling PPPoE connection with L2TP LNS address by RADIUS" on page 17
- "Example 3: Tunneling PPPoE connection with L2TP LNS address by DNS lookup" on page 19

To configure the device as a PPPoE client, see the ‘Configuring PPPoE’ section in the Point-to-Point Protocol (PPP) Feature Overview and Configuration Guide.

L2TPv2 connections

To configure an L2TPv2 LAC, the following must be configured:

Create and configure an L2TP profile (**l2tp-profile** command), give it a name and specify for the profile:

- the shared secret used to authenticate the LNS (**shared-secret** command)
- the RADIUS group to use if an LNS address is to be found by RADIUS (**server** command)
- the DNS server to query if an LNS address is to be found be DNS (**ip name-server** command)

PPPoE Access Concentrator

To configure the PPPoE AC, create a PPPoE AC instance (**pppoe-ac** command), give it a name, and specify:

- the service name for the clients to use in order to connect to this instance, if desired (the service name can be left unspecified as **any**), and whether it is to advertise the service-name (**service-name** command)
- that the PPP packets from the client will be sent to L2TP (**destination** command)
- the name of the L2TP profile to which it will send PPP packets from this client (**l2tp profile** command)
- how the LNS address is to be determined—
  DNS (**l2tp peer-address dns-lookup prefix** command)
  RADIUS (**l2tp peer-address radius-lookup group** command)
  or static (**l2tp peer-address static** command)

Example 1: Tunneling PPPoE connection with a static L2TP LNS address

Table 3: Configuration example 1: Tunneling PPPoE connections with a static L2TP LNS address

Step 1. **Configure the PPPoE service.**	

`awplus#`
`configure terminal`
	Enter Configuration mode.

`awplus(config)#`
`pppoe-ac mylittleac`
	Create and name the PPPoE AC instance and enter PPPoE AC Configuration mode.

`awplus(config-pppoe-ac)#`
`service-name any`
or
`awplus(config-pppoe-ac)#`
`service-name remote-office`
	Set the service name for the PPPoE client to connect to. Either:
	■ To set the service to allow the client to connect without checking the service-name tag from the client, use **service-name any**; or
	■ To set the PPPoE AC service to only accept connections when the service-tag in the client packets match the a particular service name, specify the **service-name**.

`awplus(config-pppoe-ac)#`
`service-name remote-office advertised`
	Specify **advertised** if the service name is to be advertised, or omit if it is not to be advertised.

`awplus(config-pppoe-ac)#`
`destination l2tp`
	Configure the PPPoE service to use L2TP for its destination.

`awplus(config-pppoe-ac)#`
`l2tp peer-address static 192.168.11.2`
	Assign a static IP address for the L2TP peer. This is the address of the LNS that will terminate the L2TP tunnel and the PPP session.

`awplus(config-pppoe-ac)#`
`l2tp profile my-l2tp-profile1`
	Specify the L2TP profile used to tunnel this PPPoE service. This is the L2TP profile created by the **l2tp-profile** command.

`awplus#` `exit`	Leave PPPoE AC Configuration mode.
Step 2. **Configure the L2TP tunnel.**	

`awplus(config)#`
`l2tp-profile my-l2tp-profile1`
	Create the L2TP profile and enter L2TP Profile Configuration mode.

`awplus(config-l2tp-profile)#`
`shared-secret oursecret`
	Configure the shared secret for this L2TP tunnel. The ISP must configure the same shared secret on the LNS for this tunnel.

`awplus#` `exit`	Leave L2TP Profile Configuration mode.
Step 3. **Configure the interface.**	

`awplus(config)#`
`int eth1`
	Enter Interface Configuration mode for the interface.

Table 3: Configuration example 1: Tunneling PPPoE connections with a static L2TP LNS address

`awplus(config-if)#`
`pppoe-ac-service mylittleac`
	Configure the interface to use the PPPoE AC service, so that PPPoE packets arriving on this interface will be processed by this PPPoE AC instance.

Example 2: Tunneling PPPoE connection with L2TP LNS address by RADIUS

Table 4: Configuration example 2: Tunneling PPPoE Connections with L2TP LNS address found by RADIUS lookup

Step 1. **Configure the PPPoE service.**	

`awplus#`
`configure terminal`
	Enter Configuration mode.

`awplus(config)#`
`pppoe-ac mylittleac`
	Create and name the PPPoE AC instance and enter PPPoE AC configuration mode.

`awplus(config-pppoe-ac)#`
`service-name any`
or
`awplus(config-pppoe-ac)#`
`service-name remote-office`
	Set the service name for the PPPoE client to connect to. Either:
	■ To set the service to allow the client to connect without checking the service-name tag from the client, use **service-name any**; or
	■ To set the PPPoE AC service to only accept connections when the service-tag in the client packets match the a particular service name, specify the **service-name**.

`awplus(config-pppoe-ac)#`
`service-name remote-office advertised`
	Specify **advertised** if the service name is to be advertised, or omit if it is not to be advertised.

`awplus(config-pppoe-ac)#`
`destination l2tp`
	Configure the PPPoE service to use L2TP for its destination.

`awplus(config-pppoe-ac)#`
`l2tp peer-address radius-lookup group my_group`
	Set the PPPoE AC service to use RADIUS lookup to get the address of the LNS. Specify the name of the RADIUS group.
	With this setting, when the PPPoE AC service uses RADIUS lookup a PPP user-name of, for instance, user@domain.com will first have the 'domain.com' portion extracted. An Access-Request packet is sent to the RADIUS server with User-Name=domain.com,User-Password='password'. An Access-Accept packet is expected back with the IPv4 address of the LNS contained in Framed-IP-Address format (e.g. 192.168.11.2).
	This is the address of the LNS that will terminate the L2TP tunnel and the PPP session.

Table 4: Configuration example 2: Tunneling PPPoE Connections with L2TP LNS address found by RADIUS lookup (continued)

awplus(config-pppoe-ac)# l2tp profile my-l2tp-profile1	Specify the L2TP profile used to tunnel this PPPoE AC service.
awplus# exit	Leave PPPoE AC configuration mode.
Step 2. **Configure the L2TP tunnel.**	
awplus(config)# l2tp-profile my-l2tp-profile1	Create an L2TP profile for the tunnel and enter the configuration mode.
awplusawplus(config-l2tp-profile)# shared-secret oursecret	Configure the shared secret for this L2TP tunnel. The ISP must configure the same shared secret on the LNS for this tunnel.
awplus# exit	Leave L2TP Profile Configuration mode.
Step 3. **Configure the LAC to use RADIUS.**	
awplus(config)# radius-server host 192.168.1.200 key testing123-1	Specify the address and key for the RADIUS server.
awplus(config)# aaa group server radius my_group	Create a RADIUS group. This is the group referred to in the **l2tp peer-address radius-lookup group** command above.
awplus(config-sg)# server 192.168.1.200	Add the address of the RADIUS server group.
awplus# exit	Leave RADIUS Server Group Configuration mode.
Step 4. **Configure the Ethernet interface.**	
awplus(config)# int eth1	Enter Interface Configuration mode for the interface.
awplus(config-if)# pppoe-ac-service mylittleac	Configure the interface to use the PPPoE AC service, so that PPPoE packets arriving on this interface will be processed by this PPPoE AC instance.

Example 3: Tunneling PPPoE connection with L2TP LNS address by DNS lookup

Table 5: Configuration example 3: Tunneling PPPoE connections with L2TP LNS address found by DNS lookup

Step 1. Configure the PPPoE service.	

`awplus#`
`configure terminal`
	Enter Configuration mode.

`awplus(config)#`
`pppoe-ac mylittleac`
	Create and name the PPPoE AC instance and enter PPPoE AC configuration mode.

`awplus(config-pppoe-ac)#`
`service-name any`
or
`awplus(config-pppoe-ac)#`
`service-name remote-office`
	Set the service name for the PPPoE client to connect to. Either:
	■ To set the service to allow the client to connect without checking the service-name tag from the client, use **service-name any**; or
	■ To set the PPPoE AC service to only accept connections when the service-tag in the client packets match the a particular service name, specify the **service-name**.

`awplus(config-pppoe-ac)#`
`service-name remote-office advertised`
	Specify **advertised** if the service name is to be advertised, or omit if it is not to be advertised.

`awplus(config-pppoe-ac)#`
`destination l2tp`
	Configure the PPPoE service to use L2TP for its destination.

`awplus(config-pppoe-ac)#`
`l2tp peer-address dns-lookup prefix lns`
	Set the PPPoE AC to use DNS lookup to get the IP address of the destination LNS.
	For a PPP user-name of, for instance, user@domain.com, the 'domain.com' portion will first be extracted, prefixed with 'lns.' to become 'lns.domain.com'. This is sent to the DNS server, which sends back the LNS address to use.
	This is the address of the LNS that will terminate the PPP session.

`awplus(config-pppoe-ac)#`
`l2tp profile my-l2tp-profile1`
	Specify the L2TP profile used to tunnel this PPPoE service.

`awplus#` `exit`	Leave PPPoE AC Configuration mode.
Step 2. Configure the L2TP tunnel.	

`awplus(config)#`
`l2tp-profile my-l2tp-profile1`
	Create and name the L2TP profile.

`awplusawplus(config-l2tp-profile)#`
`shared-secret oursecret`
	Configure the shared secret for this L2TP tunnel.
	The ISP must configure the same shared secret on the LNS for this tunnel.

Table 5: Configuration example 3: Tunneling PPPoE connections with L2TP LNS address found by DNS lookup (continued)

awplus# exit	Leave L2TP Profile Configuration mode.
Step 3. Configure the LAC to use DNS.	
awplus(config)# ip name-server 10.1.1.1	Specify the address of the DNS server from which to look up the LNS address.
Step 4. Configure the Ethernet interface.	
awplus(config)# int eth1	Enter Interface Configuration mode for the interface.
awplus(config-if)# pppoe-ac-service mylittleac	Configure the interface to use the PPPoE AC service, so that PPPoE packets arriving on this interface will be processed by this PPPoE AC instance.

Monitoring

show pppoe-ac connections

To display all PPPoE Access Concentrator (AC) connections for all routes, use the command:

```
awplus# show pppoe-ac connections
```

This will show the connections from clients and the tunnels into which these connections are being sent.

To display connected PPPoE routes for a particular PPPoE service instance named ‘pppoeservice1’, use the command:

```
awplus# show pppoe-ac pppoeservice1 connections
```

Table 6: Example output from **show pppoe-ac connections**

```
awplus# show pppoe-ac connections

PPPoE Access Concentrator Connection Status

---------------------------------------------------------------------
Route Name:                     pppoeservice-eth1
Route ID:                       29785
Source Information
  Interface:                    eth1
  Session ID:                   14204
  Service Name:                 test
  State:                        Open
  Peer MAC:                     00:00:cd:38:01:4f
Destination Information
  Type:                         L2TP
  Tunnel ID:                    11223
  Session ID:                   57309

Route Name:                     ac1-eth2
Route ID:                       34409
Source Information
  Interface:                    eth2
  Session ID:                   14108
  Service Name:                 my_isp
  State:                        Open
  Peer MAC:                     00:00:cd:38:01:4d
Destination Information
  Type                          L2TP
  Tunnel ID                     47432
  Session ID                    10056
```

show pppoe-ac statistics

To display statistics for the PPPoE access concentrator, use the command:

```
awplus# show pppoe-ac statistics
```

```
awplus# show pppoe-ac statistics

PPPoE Access Concentrator Statistics
Name                                                Value
--------------------------------------------------------------
lnsLookupSuccessfulRequests                         0
lnsLookupFailedRequests                             0
lnsLookupDnsFailures                                0
lnsLookupRadiusFailures                             0
l2tpTunnelsOpened                                   2
l2tpSessionsOpened                                  2
l2tpSessionsClosed                                  0
l2tpDnsFailures                                     0
pppoePadiReceived                                   2
pppoeInvalidPadi                                    0
pppoePadoSent                                       2
pppoePadsSent                                       2
pppoePadrReceived                                   2
pppoeInvalidPadr                                    0
pppoeResentPadr                                     0
pppoePadtReceived                                   0
pppoeInvalidPadt                                    0
pppoePadtSent                                       0
routesCreated                                       2
routesCreateFail                                    0
routesDeleted                                       0
routesDeleteFail                                    0
routesDstOpenFail                                   0
routesDestCloseFail                                 0
routesSourceCloseFail                               0
routesClosedByDest                                  0
routesClosedBySource                                0
```

show pppoe-ac config-check

To display a configuration check for all PPPoE AC service instances, use the command:

```
awplus#show pppoe-ac config-check
```

To display a configuration check for a particular PPPoE AC service instance named ‘ac1’, use the command:

```
awplus#show pppoe-ac ac1 config-check
```

This output indicates whether you have a full and valid configuration for a PPPoE instance, or which configuration still needs to be completed.

```
awplus# show pppoe-ac config-check

PPPoE Access Concentrator ac:
  Incomplete Configuration
  Required: add pppoe-ac-service to one or more interfaces
  Required: destination
  Required: service-name
  Required: l2tp peer-address
  Required: l2tp profile


PPPoE Access Concentrator ac1:
  Incomplete Configuration
  Required: add pppoe-ac-service to one or more interfaces


PPPoE Access Concentrator pppoeservice1:
  Complete Configuration
```

show running-config pppoe-ac

To display the running configuration of PPPoE Access Concentrator, use the command:

```
awplus#show running-config pppoe-ac
```

```
awplus# show running-config pppoe-ac

 pppoe-ac-service ISP-service
  service-name remote-office advertised
  ppp-auth-protocols pap
  destination l2tp
  l2tp peer-address static 192.168.11.2
  l2tp profile PUBLIC
```

show running-config l2tp-profile

To display the running configuration of L2TPv2 tunnel profiles, use the command:

```
awplus#show running-config l2tp-profile
```

```
awplus# show running-config l2tp-profile
l2tp-profile public
 version 2
 secret "my_password"
```

Debugging PPPoE Access Concentrators and L2TPv2 tunnels

To enable debugging for the PPPoE Access Concentrator, use the command:

```
awplus#debug pppoe-ac
```

To disable debugging for the PPPoE Access Concentrator, use the command:

```
awplus#no debug pppoe-ac
```

To see whether debugging of PPPoE AC is enabled or disabled, use the command:

```
awplus#show debugging pppoe ac
```

To enable debugging for L2TPv2 tunnels, use the command:

```
awplus#debug l2tp
```

To disable debugging for L2TPv2 tunnels, use the command:

```
awplus#no debug l2tp
```

To see whether debugging of L2TPv2 tunnels is enabled or disabled, use the command:

```
awplus#show debugging l2tp
```

Managed L2TPv2 Peer-to-Peer Tunnels

The implementation of L2TPv2 peer-to-peer tunnels on the AR-Series firewalls supports IPv4 or IPv6 based Layer 3 VPNs via the negotiation of the PPP LCP link-layer, and also dual-stack PPP IPCP, PPP IPV6CP Network Control Protocol (NCP) layers. Private traffic between the two peers is routed via the PPP link encapsulated by the L2TPv2 VPN. Layer 2 L2TPv2 VPNs, which involve the negotiation of PPP BCP (Bridge Control Protocol), and Layer 2 bridging of private LAN traffic onto the VPN PPP interface are not supported on the AR-Series firewalls.

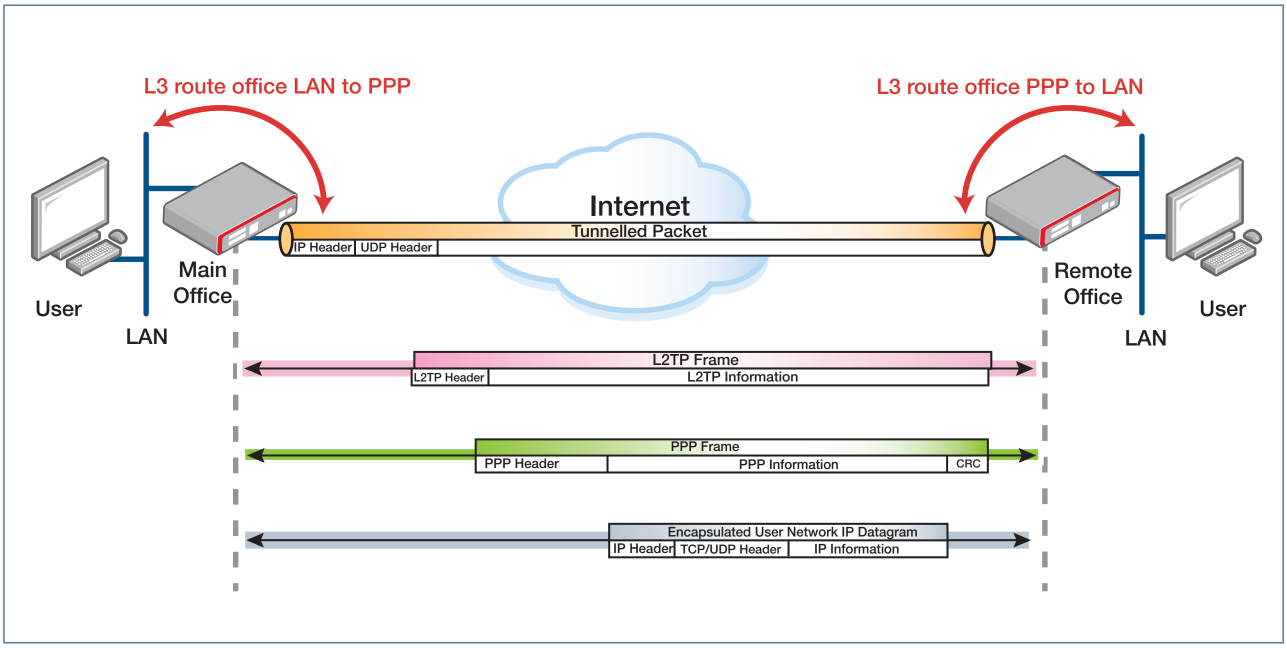
L2TPv2 can also be configured with IPsec protection to allow secure encrypted communication of the VPN traffic via the intermediate public network.

L2TP VPN peer-to-peer negotiation

The negotiation of a peer-to-peer connection is very similar to that of an LAC-LNS connection, as described in "Process of establishing an LAC-LNS connection" on page 8. The main differences are:

- Managed L2TP peer-to-peer tunnels only support one session per tunnel.
- If the initiator is configured with a remote sub-address, this is also sent in the ICRQ packet, which is compared by the recipient against the configured local sub-address.
- The PPP link is negotiated directly between the peers.

Figure 6: L2TPv2 managed peer-to-peer encapsulations: PPP frame in L2TP frame in tunneling packet



L2TP call collision arbitration

L2TPv2 is an asymmetric protocol with one node acting as the initiator and the other acting as the recipient. In L2TPv2 peer-to-peer tunnels both devices function as the initiator initially, that is, both sides of the tunnel can simultaneously attempt to initiate a connection with the remote host by sending SCCRQ (Start-Control-Connection-Request) packets. The SCCRQ packets include a tiebreaker AVP. Both devices will compare each others tiebreaker values for the tunnel during the VPN negotiation process, and the device with the lower value loses the tiebreaker and acts as the LNS. The tiebreaker loser device creates a new tunnel in LNS mode to accept the tiebreaker winner's LAC tunnel. The LAC tunnel that lost the tiebreaker process will go into an IDLE state and will retry L2TP establishment if the LNS tunnel goes down.

Details of implementation of L2TPv2 peer-to-peer tunnels on AR-Series firewalls

The implementation of the peer-to-peer tunnels shares a number of the attributes of the LAC-LNS tunnels. Significant differences are:

- Only one PPP session is supported per managed peer-to-peer tunnel.
- There is support for up to 20 L2TP peer-to-peer tunnels are supported.
- The configured remote sub-address is only sent by the initiator and checked against the recipient's configured local sub-address. This means that if the initiator's local sub-address or the recipient's remote sub-address are misconfigured, then the tunnel session could still establish, depending on which node initiated the tunnel first.
- There is no support for authentication based on the node's host-name (fully-qualified domain name), although the host-name AVP is sent automatically during tunnel negotiation, and so interoperability is possible with other vendors' L2TP implementations that authenticate based on host-name.
- In order to establish a peer-to-peer tunnel, the destination IP address or fully-qualified domain name must be configured.
- There is no support for PPP idle timer (dial-on-demand) for these tunnels.

Managed L2TPv2 Peer-to-Peer Tunnel Configuration

This section describes how to configure L2TPv2 managed peer-to-peer tunnels, and then shows the following example configurations:

- "Example 4: Basic managed peer-to-peer tunnel" on page 29
- "Example 5: Basic tunnel between AlliedWare Plus and AlliedWare" on page 30
- "Example 6: L2TP tunnel with IPsec" on page 32
- "Example 7: L2TP tunnel with IPsec between AlliedWare Plus and AlliedWare" on page 34
- "Example 8: Simultaneous L2TPv3 pseudowire and managed L2TPv2 tunnel" on page 37

The following steps describe how to configure a managed L2TP peer-to-peer tunnel. Compatible configuration settings for all these commands must be configured on both L2TP peers. These steps apply for a configuration where:

- The devices are able to communicate with each other without traversing a security device or an intermediate device running NAT.
- The device is not also using L2TPv3.

To configure a basic tunnel:

1. Create and name the L2TP tunnel (**l2tp tunnel** command):

   ```
   awplus(config)# l2tp tunnel <tunnel-name>
   ```

2. Create an associated PPP interface (**encapsulation ppp** command).

   ```
   awplus(config-l2tp-tunnel)# encapsulation ppp <ppp-int-number>
   ```

3. Specify the destination of the remote node.

   ```
   awplus(config-l2tp-tunnel)# destination [<ipv4-addr>|<ipv6-
   addr>|<domain-name>]
   ```

   When L2TP traffic egresses an interface, the IP address associated with the egress interface is used by default as the source IP for the L2TP traffic. If the destination IP address (configured in the remote peer) matches the source IP address used, then there is no need to specify an L2TP source. The destination IP address alone is sufficient.

4. However, if necessary, configure the source interface. This allows L2TP traffic from the remote node to be addressed to an interface other than the local egress interface. For example, you may wish the remote node to send its L2TP traffic to the loopback address of the current node.

   ```
   awplus(config-l2tp-tunnel)# source [<ipv4-addr>|<ipv6-addr>|
   <interface>]
   ```

Other options that can be configured are:

- **Remote subaddress**

  To allow interoperability with AlliedWare's implementation L2TPv2 peer-to-peer tunnels, a remote subaddress that matches the associated call on the AlliedWare device must be configured (**remote-subaddress** command). Use the command:

  ```
  awplus(config-l2tp-tunnel)# local-subaddress [<remote-
  subaddress-of-remote-node>
  ```

  Note that configuring a remote subaddress could lead to interoperability issues with third party vendor equipment that instead use alternative methods of peer identification, such as the host-name AVP.

- **IPsec**

  To configure IPsec encryption for L2TPv2 peer-to-peer tunnels, use the command:

  ```
  awplus(config-l2tp-tunnel)#  protection ipsec [profile
  <profile-name>] [local-id <id>] [remote-id <id>]
  ```

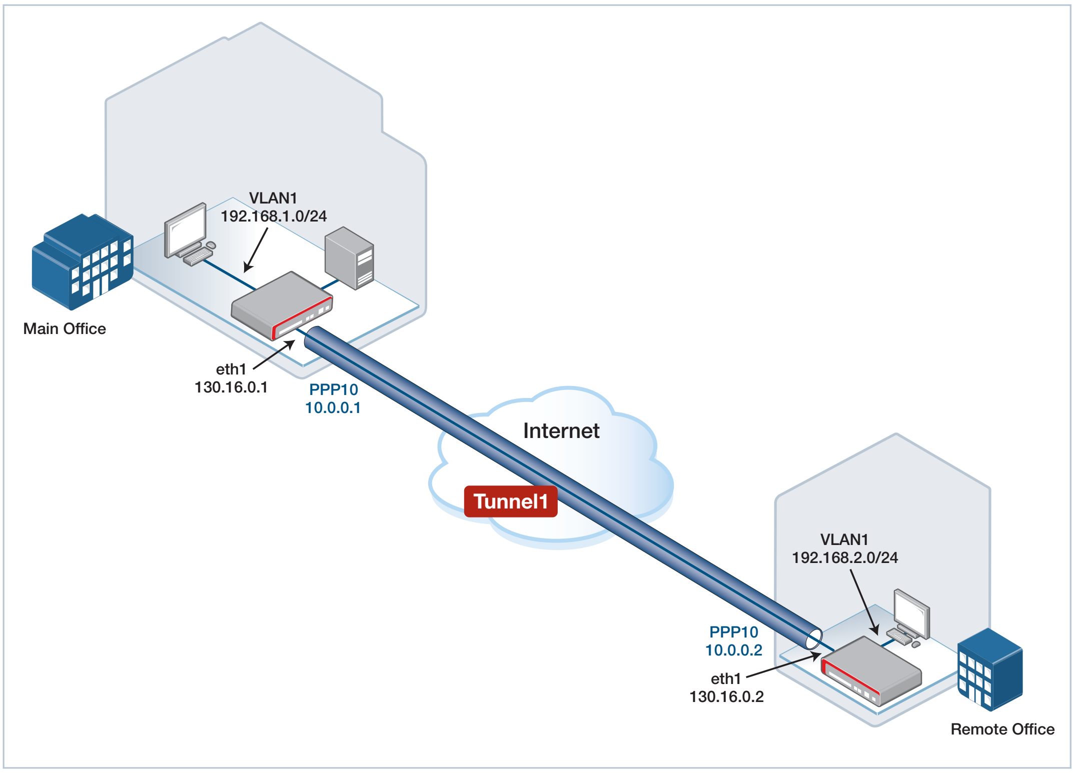
- **Authentication**

  To configure a shared secret to be used for authentication between nodes, use the command:

  ```
  awplus(config-l2tp-tunnel)# shared-secret <password>
  ```

Example 4: Basic managed peer-to-peer tunnel

This is a basic managed L2TP peer-to-peer tunnel with configurations for the AR-Series firewalls at each end of the tunnel. In this example, VLAN1 is configured as the private LAN network in both Main and Remote offices. Eth1 is used as the WAN interface, and PPP 10 is used by L2TP. The private network traffic flowing between main and remote office LANs is routed over the PPP interface, which is in turn encapsulated inside L2TP headers, which is in turn transmitted via the Eth WAN.



Example 4: Main Office

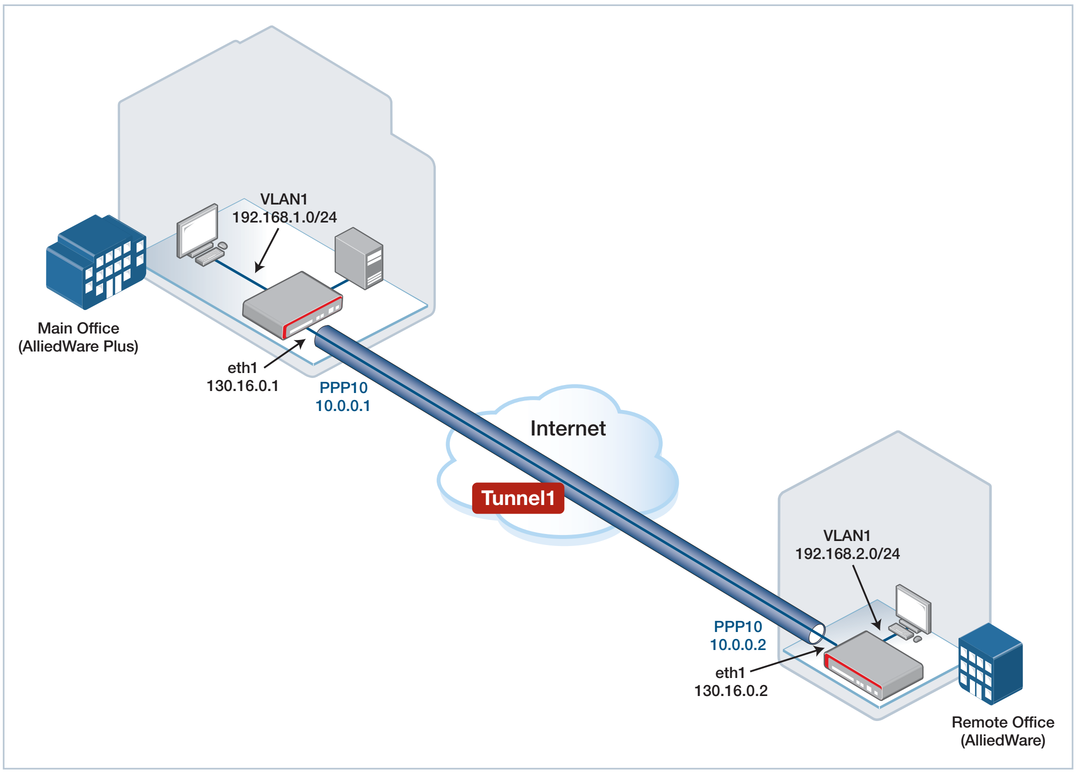
```
interface eth1
 ip address 130.16.0.1/30
!
interface vlan1
 ip address 192.168.1.254/24
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 destination 130.16.0.2
!
interface ppp10
 ip addr 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
```

Example 4: Remote Office

```
interface eth1
 ip address 130.16.0.2/30
!
interface vlan1
 ip address 192.168.2.254/24
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 destination 130.16.0.1
!
interface ppp10
 ip addr 10.0.0.2/30
!
ip route 192.168.1.0/24 10.0.0.1
```

Example 5: Basic tunnel between AlliedWare Plus and AlliedWare

This is a basic managed L2TP peer-to-peer tunnel with configurations for an AR-Series firewall at one end of the tunnel (Main office) and an older device, running AlliedWare, at the other (Remote office), similar to the previous example.



Example 5: Main Office—AlliedWare Plus

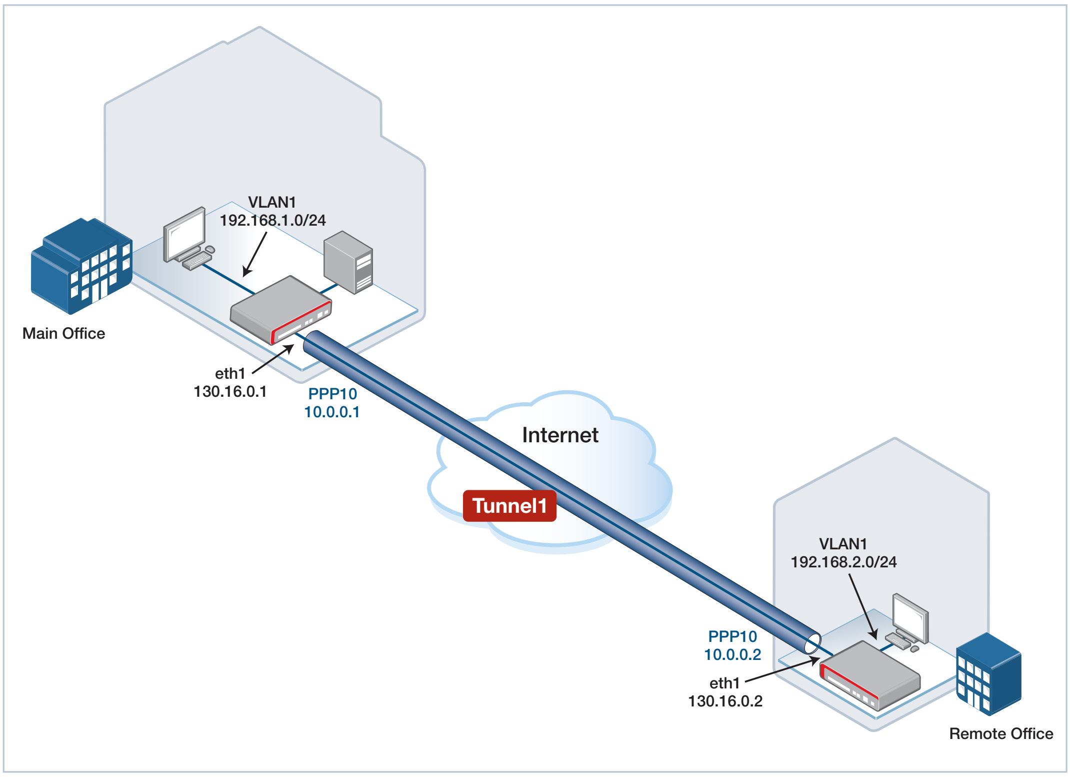
```
interface eth1
 ip address 130.16.0.1/30
!
interface vlan1
 ip address 192.168.1.254/24
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 destination 130.16.0.2
 remote-subaddress remote_office
 local-subaddress main_office
!
interface ppp10
 ip addr 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
```

Example 5:Remote Office—AlliedWare

```
enable l2tp
enable l2tp server=both
add l2tp call="remote_office" remotecall="main_office" ip=130.16.0.1
  ty=virtual prec=in
create ppp=10 over=tnl-remote_office
set ppp=10 over=tnl-remote_office lqr=off echo=on
add ip int=vlan1 ip=192.168.2.254 mask=255.255.255.0
add ip int=ppp10 ip=10.0.0.2 mask=255.255.255.252
add ip int=eth1 ip=130.16.0.2 mask=255.255.255.252
add ip rou=192.168.1.0 mask=255.255.255.0 int=ppp10 next=10.0.0.1
```

Example 6: L2TP tunnel with IPsec

This examples builds on the previous examples, and shows how to configure managed L2TP VPN that is protected using IPsec encryption. Once the pre-shared key is configured, and the Tunnel is protected by IPsec, the internal default IPsec profile is automatically used for establishing the encrypted VPN.



Example 6: Main Office—AlliedWare Plus

```
crypto isakmp key samplekey address 130.16.0.2
!
interface eth1
 ip address 130.16.0.1/30
!
interface vlan1
 ip address 192.168.1.254/24
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 destination 130.16.0.2
 source 130.16.0.1
 protection ipsec
!
interface ppp10
 ip address 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
```

Example 6: Remote Office—AlliedWare Plus

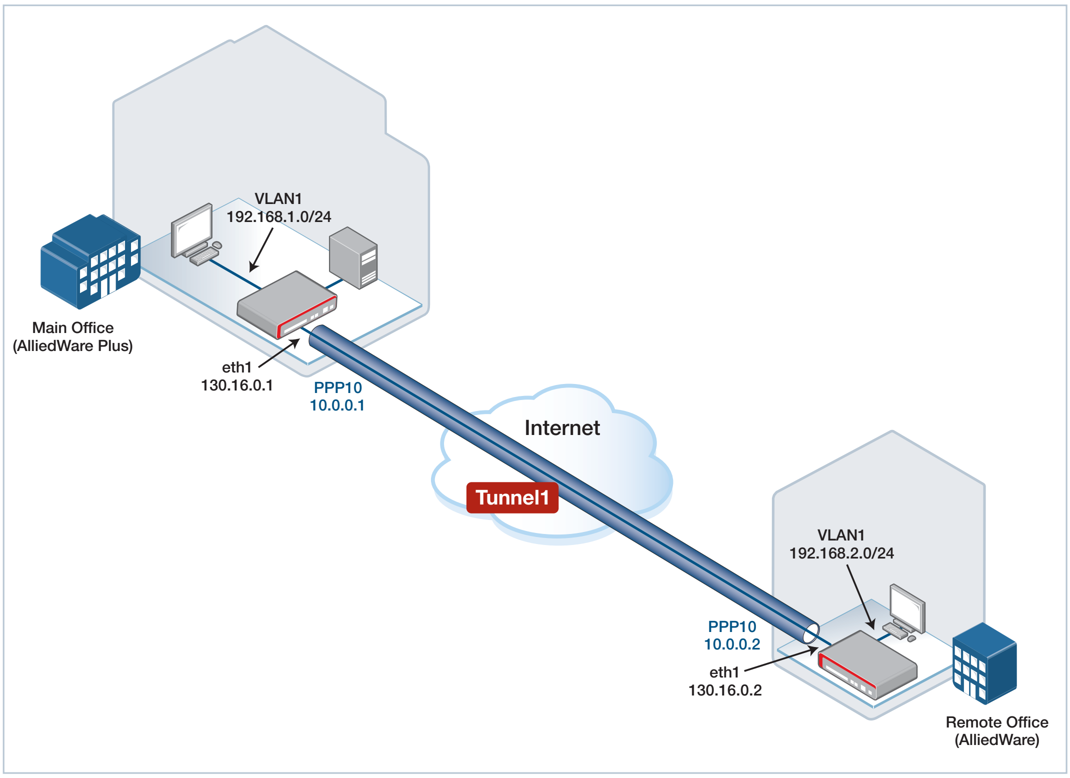
```
crypto isakmp key samplekey address 130.16.0.1
!
interface eth1
 ip address 130.16.0.2/30
!
interface vlan1
 ip address 192.168.2.254/24
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 destination 130.16.0.1
 source 130.16.0.2
 protection ipsec
!
interface ppp10
 ip address 10.0.0.2/30
!
ip route 192.168.1.0/24 10.0.0.1
```

Example 7: L2TP tunnel with IPsec between AlliedWare Plus and AlliedWare

This example shows how to configure a managed L2TP peer-to-peer tunnel with IPsec protection operating between an AR-Series firewall at Main office, and an older device, running AlliedWare, at Remote office.

The AR-Series firewall is configured with customized ISAKMP and IPsec profiles, with legacy cryptographic options to allow the ISAKMP and IPsec VPN security associations to negotiate successfully with the AlliedWare device.

Additionally, in the device running AlliedWare, IPsec is configured to use transport mode, and the local and remote IPsec address selectors are configured to match the WAN IP addresses allocated to each device. This is because the traffic to be encrypted is encapsulated inside L2TP and the L2TP traffic is sourced from the WAN IP address used in each device.



Example 7: Main Office—AlliedWare Plus

```
!
crypto ipsec profile AW
 lifetime seconds 300
 pfs 2
 transform 1 protocol esp integrity SHA1 encryption AES128
!
crypto isakmp profile AW
 version 1 mode main
 transform 1 integrity SHA1 encryption AES128 group 2
!
crypto isakmp key samplekey address 130.16.0.2
!
crypto isakmp peer address 130.16.0.2 profile AW
!
l2tp tunnel 1
 version 2
 ip-version 4
 encapsulation ppp 10
 source 130.16.0.1
 destination 130.16.0.2
 local-subaddress main_office
 remote-subaddress remote_office
 protection ipsec
 protection profile AW
!
interface eth1
 ip address 130.16.0.1/30
!
interface vlan1
 ip address 192.168.1.254/24
!
interface ppp10
 ip address 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
```

Example 7: Remote Office—AlliedWare

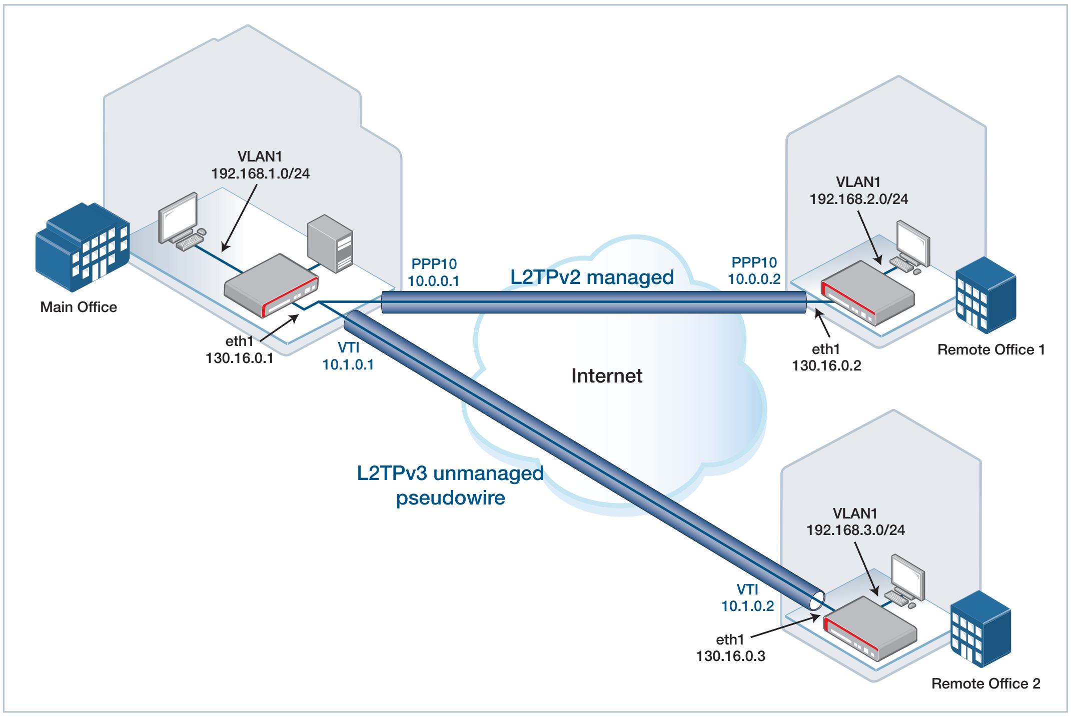
```
enable l2tp
enable l2tp server=both
add l2tp call="remote_office" rem="main_office" ip=130.16.0.1
ty=virtual prec=in

create ppp=10 idle=9999 over=tnl-remote_office
set ppp=10 over=tnl-remote_office lqr=off echo=on

enable ip
add ip int=eth0 ip=130.16.0.2 mask=255.255.255.252
add ip int=ppp10 ip=10.0.0.2 mask=255.255.255.252
add ip int=vlan1 ip=192.168.2.254 mask=255.255.255.0
add ip rou=192.168.1.0 mask=255.255.255.0 int=ppp10 next=10.0.0.1

create enco key=1 type=general value=samplekey
create ipsec sas=1 key=isakmp prot=esp enc=aes128 hasha=sha
set ipsec sas=1 mod=transport
create ipsec bund=1 key=isakmp string="1" expirys=300
create ipsec pol="office_vpn_isakmp" int=eth0 ac=permit
set ipsec pol="office_vpn_isakmp" lp=500 rp=500
create ipsec pol="office_vpn_ipsec" int=eth0 ac=ipsec key=isakmp bund=1
peer=130.16.0.1
set ipsec pol="office_vpn_ipsec" lad=130.16.0.2 rad=130.16.0.1
set ipsec pol="office_vpn_ipsec" usepfsk=TRUE gro=2
enable ipsec

create isakmp pol="L2TP" pe=130.16.0.1 enc=aes128 key=1
set isakmp pol="L2TP" gro=2
set isakmp pol="L2TP" sendd=true
enable isakmp
```

Example 8: Simultaneous L2TPv3 pseudowire and managed L2TPv2 tunnel

Both unmanaged L2TPv3 static pseudowires and managed L2TPv2 tunnels attempt to use UDP port 1701 by default. In order to use both tunnel types simultaneously on the same device, and allow the L2TPv3 and L2TPv2 traffic to be distinguishable, you must change the UDP port that the unmanaged L2TPv3 pseudowires use via the **l2tp unmanaged port <*port-number*>** command. Managed L2TPv2 tunnels will continue to use the default UDP port 1701.

Any L2TPv3 pseudowires already configured must first be removed before changing the port.

The following shows a main office device peering to two remote offices. The connection from the Main office to Remote Office 1 uses default UDP port number 1701 for the managed L2TPv2 VPN that encapsulates PPP10. The connection from the Main office to Remote Office 2 is “port-shifted” to use UDP port 1702 for the unmanaged L2TPv3 pseudowire that is terminated by the L2TPv3 mode Virtual Tunnel Interface (VTI).



Example 8: Main Office—managed and unmanaged tunnels

```
l2tp unmanaged port 1702
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 source 130.16.0.1
 destination 130.16.0.2
!
interface eth1
 ip address 130.16.0.1/29
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel1
 description “unmanaged L2TPv3 VTI”
 tunnel source 130.16.0.1
 tunnel destination 130.16.0.3
 tunnel mode l2tp v3
 ip address 10.1.0.1/30
!
interface ppp10
 ip address 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
ip route 192.168.3.0/24 10.1.0.2
```

Example 8: Remote Office 1—managed tunnel

```
!
l2tp tunnel tunnel1
 encapsulation ppp 10
 source 130.16.0.2
 destination 130.16.0.1
!
interface eth1
 ip address 130.16.0.2/29
!
interface vlan1
 ip address 192.168.2.254/24
!
interface ppp10
 ip address 10.0.0.2/30
!
ip route 192.168.1.0/24 10.0.0.1
```

Example 8: Remote Office 2—unmanaged tunnel

```
l2tp unmanaged port 1702
!
interface eth1
 ip address 130.16.0.3/29
!
interface vlan1
 ip address 192.168.3.254/24
!
interface tunnel1
 description “unmanaged L2TPv3 VTI”
 tunnel source 130.16.0.3
 tunnel destination 130.16.0.1
 tunnel mode l2tp v3
 ip address 10.1.0.2/30
!
ip route 192.168.1.0/24 10.1.0.1
```

If an attempt is made to configure both managed L2TPv2 tunnels and unmanaged static L2TP pseudowires without changing the port then an error message will be displayed as follows:

```
LAC(config-if)#tunnel mode l2tp v3
% Error: L2TPv3 pseudowires and managed L2TPv2 tunnels cannot share UDP
port 1701.
Reconfigure pseudowires to use an alternative UDP port.
```

Monitoring

show l2tp tunnel config-check command

To check the configuration of all tunnels for errors or missing required configuration, use the command:

```
awplus# show l2tp tunnel config-check
```

To check the configuration of a particular specified tunnel, use the command:

```
awplus# show l2tp tunnel <tunnel-name> config-check
```

```
awplus#show l2tp tunnel config-check

L2TP Tunnel tunnel1:
  Complete Configuration


L2TP Tunnel tunnel2:
  Incomplete Configuration
  Missing tunnel name or tunnel name mismatch
  Required: PPP interface
  Required: destination
  Required: IP version
  Destination IP address does not match the configured IP version
  Source IP address does not match the configured IP version
  Source IP address required when IPSec protection on
```

An L2TP tunnel must have complete configuration to appear in the rest of the show commands.

show l2tp tunnel command

To display basic details of all configured L2TP tunnels, use the command:

```
awplus# show l2tp tunnel
```

To view basic details of a specified L2TP tunnel, use the command:

```
awplus# show l2tp tunnel <tunnel-name>
```

```
awplus#show l2tp tunnel

L2TP Tunnel Information

---------------------------------------------------------------------
Tunnel ID:                      11008
Tunnel Name:                    tunnel1
Local IP Address:               130.16.0.1
Remote IP Address:              130.16.1.2
State:                          ESTABLISHED
Created At:                     Jun  1 12:25:25 2016
Tunnel Mode:                    LAC
Remote Tunnel ID:               33363
Remote Host Name:               AWP-1
Remote Vendor Name:             Allied Telesis International
Locol UDP Port:                 1701
Remote UDP Port:                1701
Hello Timeout:                  60
Retry Timeout:                  1
Idle Timeout:                   0
Establish Timeout:              60
```

Note that if an established tunnel is in LNS mode, then there will be a corresponding idle tunnel in LAC mode. This is because both devices send out L2TP requests as a LAC but one device will win the tiebreaker. The loser will become the LNS and accept the LAC’s request. The device with the LNS tunnel will still keep the LAC tunnel in existence.

To display more detailed information of all configured L2TP tunnels, use the command:

```
awplus# show l2tp tunnel detail
```

```
awplus#show l2tp tunnel detail

L2TP Tunnel Information details

Tunnel 11008, from 130.16.0.1 to 130.16.1.2:-
  state: ESTABLISHED
  created at:  Jun  1 12:25:25 2016
  administrative name: 'tunnel1'
  created by admin: YES, tunnel mode: LAC, persist: YES
  peer tunnel id: 33363, host name: AWP-1
  UDP ports: local 1701, peer 1701
  authorization mode: NONE, hide AVPs: OFF
  session limit: 0, session count: 1
  tunnel profile: tunnel_profile_tunnel1, session profile:
    session_profile_tunnel1, peer profile: peer_profile_tunnel1
  hello timeout: 60, retry timeout: 1, idle timeout: 0
  establish timeout: 60
  persist pend timeout: 60
  rx window size: 10, tx window size: 10, max retries: 5
  use udp checksums: ON
  do pmtu discovery: OFF, mtu: 1460
  tos: inherit
  framing capability: SYNC ASYNC, bearer capability: DIGITAL ANALOG
  use tiebreaker: ON
  tiebreaker: 88 95 39 14 88 9c 44 e4
  interopability flags: 128
  trace flags: PROTOCOL FSM API AVP FUNC XPRT DATA PPP SYSTEM


  Status:-
    peer vendor name: Allied Telesis International
    peer protocol version: 1.0, firmware 385
    peer framing capability: SYNC ASYNC
    peer bearer capability: DIGITAL ANALOG
    peer rx window size: 10
  Transport status:-
    ns/nr: 39/8, peer 39/8
    cwnd: 10, ssthresh: 10, congpkt_acc: 0
  Transport statistics:-
    out-of-sequence control/data discards: 0/0
    zlbs tx/txfail/rx: 6/0/37
    retransmits: 0, duplicate pkt discards: 0, data pkt discards: 0
    hellos tx/txfail/rx: 35/0/5
    control rx packets: 45, rx bytes: 725
    control tx packets: 45, tx bytes: 1024
    data rx packets: 3621, rx bytes: 5272542, rx errors: 0
    data tx packets: 3621, tx bytes: 5301510, tx errors: 0
  memory usage: 2065 bytes
  Events:-
    12:25:25 OPEN_REQ in state IDLE, new state WAITCTLREPLY
    12:25:26 SCCRP_ACCEPT in state WAITCTLREPLY, new state ESTABLISHED
```

show l2tp session command

To display basic details of the sessions on all configured L2TP tunnels, use the command:

```
awplus# show l2tp session
```

```
awplus#show l2tp session

L2TP Session Information

---------------------------------------------------------------------
Session ID:                     6781
Tunnel ID:                      11008
State:                          ESTABLISHED
Created At:                     Jun  1 12:25:25 2016
Interface Name:                 ppp10
Remote Session ID:              23809
Establish Timeout:              120
```

To view more detailed information of the sessions of all configured L2TP tunnels, use the command:

```
awplus# show l2tp session detail
```

```
awplus#show l2tp session detail

L2TP Session Information details

Session 6781 on tunnel 11008:-
  type: LAC Incoming Call, state: ESTABLISHED
  created at:  Jun  1 12:25:25 2016
  administrative name: session_tunnel1
  interface name: ppp10
  created by admin: YES, peer session id: 23809
  session profile name: session_profile_tunnel1
  ppp profile name: ppp_profile_ppp10
  use data sequence numbers: OFF
  establish timeout: 120
  trace flags: PROTOCOL FSM API AVP FUNC XPRT DATA PPP SYSTEM
  framing types: SYNC ASYNC
  bearer types: DIGITAL ANALOG
  call serial number: 1
  sub address: 'office1'
Status:-
  data rx packets: 3621, rx bytes: 5272542, rx errors: 0
  data tx packets: 3621, tx bytes: 5301510, tx errors: 0
  Peer configuration data:-
    framing types: NONE
    bearer types: NONE
  memory usage: 1904 bytes
  Events:-
    12:25:25 INCALL_IND in state IDLE, new state WAITTUNNEL
    12:25:27 TUNNEL_OPEN_IND in state WAITTUNNEL, new state WAITREPLY
    12:25:27 ICRP_ACCEPT in state WAITREPLY, new state ESTABLISHED
```

show interface command

The L2TP tunnel is not an interface. To display the state of the PPP interface that the L2TP session is using, use the command:

```
awplus# show interface <ppp-interface>
```

```
awplus#show int ppp10
Interface ppp10
  Link is UP, administrative state is UP
  Hardware is PPP
  IPv4 address 10.0.0.1/30 point-to-point 10.0.0.2
  index 16778241 metric 1 mtu 1492
  <UP,POINT-TO-POINT,RUNNING,NOARP,MULTICAST>
  PPP is running, underlying interface (index 0) is up
  SNMP link-status traps: Disabled
    input packets 3619, bytes 5236292, dropped 0, multicast packets 0
    output packets 3619, bytes 5236292, multicast packets 0 broadcast
      packets 0
  Time since last state change: 0 days 00:39:22
```

Allied Telesis™ NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com