Technical Guide

Point-to-Point Protocol (PPP)

Feature Overview and Configuration Guide

Introduction

This guide describes AlliedWare Plus™ Point-to-Point (PPP) and its configuration.

PPP specified in RFC 1661, is a protocol used to establish a direct connection between two nodes via a WAN or LAN. It provides a standard method for transporting multi-protocol datagrams over point-to-point links. PPP protocol encapsulation provides multiplexing of different Network Layer protocols simultaneously over the same link. PPP is the most common protocol for linking a host to an ISP.

Products and software version that apply to this guide

Most features described in this document are supported from AlliedWare Plus™ software version 5.4.5 or later. The following features became available in later releases:

- From software version 5.5.0-1.4 the PPPoE relay feature is supported, see "PPPoE relay" on page 23

However, implementation varies between products. To see whether a product supports a feature or command, 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.

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

Contents

Overview

PPP was developed by the Internet Engineering Task Force (IETF) as a means of transmitting data containing more than one network protocol over the same point-to-point link in a standard, vendor-independent way.

PPP provides direct connections over synchronous and asynchronous circuits. PPP works with several network layer protocols, such as IP and IPv6. PPP also has built-in security mechanisms such as PAP (Password Authentication Protocol), CHAP (Challenge Authentication Handshake Protocol), and EAP (Extensible Authentication Protocol).

The PPP protocol consists of the following main components:

- A method for encapsulating datagrams over serial or other underlying links. HDLC (High Level Data Link Control), L2TP (Layer 2 Tunneling Protocol), and PPPoE (Point-to-Point Protocol over Ethernet) provide such protocols.
- A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
- A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols. PPP enables the simultaneous use of multiple network layer protocols. A common NCP is Internet Protocol Control Protocol (IPCP).

The method that PPP uses to carry network traffic is to open a link with a short exchange of frames. Once the link is open, network traffic is carried with very little overhead. Traffic is transmitted as a series of unnumbered information frames, meaning that no data link acknowledgments are required and no retransmissions are sent. Once the link is established, PPP acts as a straight data pipe for the upper layer protocols that it encapsulates.

Architecture

The PPP and OSI protocols share the same physical layer, but PPP distributes the functions of LCP and NCP differently. PPP operates across any DTE/DCE interface. The only requirement imposed by PPP is a duplex circuit, either dedicated or switched, that can operate in either an asynchronous or synchronous bit-serial mode and is transparent to PPP link layer frames. PPP does not impose any restrictions regarding transmission rate other than those imposed by the particular DTE/DCE lower layer interface in use.

Most of the work done by PPP is at the data link and network layers by the LCP and NCPs. The LCP sets up the PPP connection and its parameters, the NCPs handle higher layer protocol configurations, and the LCP terminates (closes) the PPP connection.

Encapsulation

PPP over Ethernet uses 8 bytes of the Ethernet frame as overhead, reducing the maximum size of IP packets that can be transmitted without fragmenting from 1500 bytes to 1492 bytes.

In the PPP encapsulation:

- The first four bytes of a PPP frame comprise a 1 octet address field that is always set to 0xFF, a 1-octet control field that is always set to 0x03 (“unnumbered information”), and a 2-octet protocol field.
- The receiving device interprets the data following the address and control fields depending on the frame's encapsulation.

The Link Control Protocol (LCP) brings up the PPP link before any other protocols can begin transmission. Each protocol carried over PPP has an associated Network Control Protocol (NCP) that negotiates options for the protocol and brings up the link for that protocol.

Control protocols

Control protocols are those run by PPP to enable a link connecting two stations to carry specific upper layer protocol types. The Link Control Protocol (LCP) must run before any other control protocol in order for the link to operate.

The local and remote stations negotiate the configuration options to be used on the link. To initiate the negotiation process, the local station sends a configure request frame, containing configuration options. The remote station responds with a frame confirming that the options are okay, suggesting different options or rejecting the options. This exchange takes place in both directions and when a station has sent and received an acknowledge packet the link layer is declared open.

Once LCP has opened the link layer, an appropriate method of authentication can be applied. When authentication has been completed successfully, or if no authentication is required, a Network Control Protocol (NCP) then runs for each network layer protocol using the link. The NCPs operate in a similar way to the LCP, negotiating configuration options specific to the network layer protocol. No

NCPs can use the PPP link until the LCP has opened the link, and no data packets can be exchanged unless the appropriate NCP is open.

Control protocols consist of states, events, and frame exchanges. Events cause link state changes. Two important events are open and close. These can either be caused by a management command or initiated internally, such as when the device powers up or an underlying link state change occurs. An open event causes the control protocol to try to establish a link; a close event terminates a link. Other events are the hardware becoming available (up) or unavailable (down), timeouts, and the arrival of frames.

Link Control Protocol (LCP) layer

The LCP layer is the working part of PPP. Architecturally, LCP sits on top of the physical layer and plays a role in establishing, configuring, and testing the data-link connection. The LCP establishes the point-to-point link. It also negotiates and sets up control options on the WAN link, which are handled by the NCPs. LCP also terminates the point-to-point connection.

The LCP provides automatic configuration of the interfaces at each end, including:

- handling varying limits on packet size
- detecting common misconfiguration errors
- terminating the link
- determining whether the link is functioning correctly

PPP uses LCP to determine the encapsulation formats as soon as the link is established.

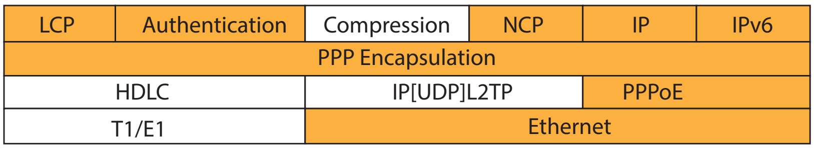
Network Control Protocol (NCP) layer

PPP permits multiple network layer protocols to operate on the same communications link. For every network layer protocol, PPP uses a separate NCP. For example, IP uses the IP Control Protocol (IPCP) and IPv6 uses IP6CP.

NCPs include functional fields containing standardized codes to indicate the network layer protocol that PPP encapsulates. Each NCP manages the specific needs of its respective network layer protocols. The various NCP components encapsulate and negotiate options for multiple network layer protocols.

PPP implementation on AlliedWare Plus

PPP on AlliedWare Plus is supported via PPP over Ethernet interfaces as a PPPoE client. Currently, no compression is supported. PPP on AlliedWare Plus supports IPv4 and IPv6.



<table>
<tr><td>LCP</td><td>Authentication</td><td>Compression</td><td>NCP</td><td>IP</td><td>IPv6</td></tr>
<tr><td colspan="6">PPP Encapsulation</td></tr>
<tr><td colspan="2">HDLC</td><td colspan="2">IP[UDP]L2TP</td><td colspan="2">PPPoE</td></tr>
<tr><td colspan="2">T1/E1</td><td colspan="4">Ethernet</td></tr>
</table>

This simplified diagram above shows the Network layer protocol using PPP. Note that only the yellow sections of this diagram are currently supported. See the items below related to this diagram:

- Ethernet— PPP over Ethernet interfaces supported.
- LCP—Link Control Protocol, establishes the connection and negotiates connection options
- NCP—PPP allows multiple protocol datagrams encapsulation on the same link. For every network protocol used, a separate Network Control Protocol (NCP) is provided in order to negotiate options for the multiple datagram network layer protocols.
  Supported NCPs are: IPCP, to configure IPv4 and IPv6CP, to configure IPv6.

PPP feature functionality

This table lists the PPP feature functionality supported:

FEATURE FUNCTIONALITY SUPPORTED
Support dynamic IPv4 addressing on a PPP interface
Support static IPv4 addressing on a PPP interface
Support constrained IPv4 addressing on a PPP interface
Apply received DNS configuration to local device
Support Primary and Secondary DNS IPCP options
LCP echo is disabled by default (but can be enabled)
LCP echo time is configurable in seconds
Ability to respond automatically to echo requests from peer
Magic Number LCP option is enabled for looped-back detection
Ability to configure static IPv6 addresses on PPP interface
IPv6 Link local address of a PPP interface can be statically configured using prefix FE80, or can be dynamically constructed based on EUI64 identifiers derived from EUI48 MAC address of the default VLAN on PPP interface
IPv6 RA commands to carry out associated actions on PPP interfaces can be configured
IPv6 RA messages received are processed and autoconf addresses are installed
Default route over PPP interface supported
Ability to enable/disable PPP debugging in conjunction with terminal monitor
PAP authentication support
CHAP authentication support
EAP authentication support (with MD5 or SRP - SHA1)
Ability to check options configured and outcome of negotiation (show interface ppp)
MRU LCP option is sent with an appropriate value if the MRU is set to a non-default value
MRU LCP option is received and action is taken, such as fragmentation
PPP over PPPoE client connections
PPPoE relay

PPP session establishment

Establishing a PPP session is a two stage process comprising:

1. Link establishment and configuration negotiation—before PPP exchanges any network layer datagrams (for example, IP), the LCP must first open the connection and negotiate configuration options. This phase is complete when the receiving router sends a configuration-acknowledgment frame back to the router initiating the connection.
2. Network layer protocol configuration negotiation—after the LCP has finished the link quality determination phase, the appropriate NCP can separately configure the Network layer protocols, and bring them up. If the LCP closes the link, it informs the network layer protocols so that they can take appropriate action.

The link remains configured for communications until explicit LCP frames close it, or until some external event occurs, such as an inactivity timer expires or a user intervenes. The LCP can terminate the link at any time. This is usually done when one of the routers requests termination, but can happen because of a physical event, such as the loss of a carrier or the expiration of an idle-period timer.

Establishing a PPP link with LCP

LCP operation

LCP operation includes provisions for PPP link establishment, maintenance, and termination. LCP operation uses three classes of LCP frames to accomplish the work of each of the LCP phases:

- Link-establishment frames establish and configure a link (Configure-Request, Configure-Ack, Configure-Nak, and Configure-Reject)
- Link-maintenance frames manage and debug a link (Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply, and Discard-Request)
- Link-termination frames terminate a link (Terminate-Request and Terminate-Ack)

The first phase of LCP operation is link establishment. This phase must complete successfully, before any Network layer packets can be exchanged. During link establishment, the LCP opens the connection and negotiates the configuration parameters.

The link establishment process starts with the initiating device sending a Configure-Request frame to the responder. The Configure-Request frame includes a variable number of configuration options needed to set up on the link. In other words, the initiator has sent a “wish list” to the responder.

The initiator's wish list includes options for how it wants the link created, including protocol or authentication parameters. The responder processes the wish list, and if it is acceptable responds with a Configure-Ack message. After receiving the Configure-Ack message, the process moves on to the authentication stage.

If the options are not acceptable or not recognized the responder sends a Configure-Nak or Configure-Reject. If a Configure-Ack is received, the operation of the link is handed over to the NCP. If either a Configure-Nak or Configure-Reject message is sent to the requester, the link is not established. If the negotiation fails, the initiator needs to restart the process with a new or a reduced set of options.

During link maintenance, LCP can use messages to provide feedback and test the link.

Code-Reject and Protocol-Reject—frames provide feedback when one device receives an invalid frame due to an unrecognized LCP frame type or a bad protocol identifier. If a packet that cannot be interpreted is received from the peer, a Code-Reject packet is sent in response.

Echo-Request, Echo-Reply, and Discard-Request—frames can be used to test the link.

After the transfer of data at the Network layer completes, LCP terminates the link. NCP only terminates the Network layer and NCP link. The link remains open until LCP terminates it. If LCP terminates the link before NCP, then the NCP session is also terminated.

PPP can terminate the link at any time. This might happen because of authentication failure or the administrative closing of the link. The LCP closes the link by exchanging Terminate packets. The device initiating the shutdown sends a Terminate-Request message. The other device replies with a Terminate-Ack. A termination request indicates that the device sending it needs to close the link. When the link is closing, PPP informs the Network layer protocols so that they may take appropriate action.

PPP configuration options

PPP can be configured to support Authentication using either PAP, CHAP, or EAP.

To negotiate Authentication, the LCP link-establishment frames contain Option information in the Data field of the LCP frame. If an Authentication option is not included in an LCP frame, the default value for that Authentication option is assumed. This phase is complete when a configuration acknowledgment frame has been sent and received.

NCP processing

After the link has been initiated, the LCP passes control to the appropriate NCP. Although initially designed for IP datagrams, PPP can carry data from many types of Network layer protocols by using a modular approach in its implementation. It can also carry two or more Layer 3 protocols simultaneously. Its modular model allows the LCP to set up the link and then hand the details of a network protocol to a specific NCP. Each network protocol has a corresponding NCP. NCPs use the same packet format as the LCPs.

After the LCP has configured and authenticated the basic link, the appropriate NCP is invoked to complete the specific configuration of the Network layer protocol being used. When the NCP has successfully configured the Network layer protocol, the network protocol is in the open state on the established LCP link. At this point, PPP can carry the corresponding Network layer protocol packets.

IPCP example

The NCP for IPv4 is the Internet Protocol Control Protocol (IPCP). After LCP has established the link, the routers exchange IPCP messages, negotiating options specific to the protocol. IPCP is responsible for configuring, enabling, and disabling the IP modules on both ends of the link.

IPCP negotiates IP addresses and DNS options. It allows the initiating device to specify an IP address to use for routing IP over the PPP link, or to request an IP address for the responder. See the **ip address negotiated** command, the **peer default ip address** command, and the **ppp ipcp dns** command, for detailed PPP IPCP command descriptions and command examples to specify an IP address for a PPP link.

When the NCP process is complete, the link goes into the open state and LCP takes over again. Link traffic consists of any possible combination of LCP, NCP, and Network layer protocol packets. LCP messages can then be used to monitor, manage, or debug the link.

LCP configuration options

PPP may include the following LCP options:

- Authentication—Peer routers exchange authentication messages. Three authentication choices are Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Extensible Authentication Protocol (EAP).
- Error detection—Identifies fault conditions. The LCP Keepalive and Magic Number options help ensure a reliable, loop-free data link. The Magic Number field helps in detecting links that are in a looped-back condition. Until the Magic-Number Configuration Option has been successfully negotiated, the Magic-Number must be transmitted as zero. Magic numbers are generated randomly at each end of the connection.
- LCP Keepalive messages can be sent periodically across the link. If several LCP Keepalive responses fail to be received, then LCP can detect the connection to the peer has failed and automatically initiate the PPP link closure.

Peer neighbor routes

AlliedWare Plus creates neighbor routes on a PPP interface by default after PPP IPCP negotiation has completed. See the product’s Command Reference for relevant commands:

- **peer neighbor-route**—the purpose of this command is to re-enable creation of neighbor routes after neighbor routes have been disabled with the no form of this command.
- **no peer neighbor-route**—the purpose of this command is to disable neighbor routes.

PPP link configuration

This example shows you how to configure a PPP link with an IP address on an Ethernet interface.

Enabling PPP encapsulation on an Ethernet interface

To set PPP as the encapsulation method used by an Ethernet interface, use the **encapsulation ppp** command from the Interface Configuration mode. The following example enables PPP encapsulation on Ethernet interface eth1:

```
awplus#configure terminal
awplus(config)#interface eth1
awplus(config-if)#encapsulation ppp 0
```

See the **encapsulation ppp** command for a detailed command description and command examples.

Obtaining an IP address for a PPP Link

The following example obtains an IP address for the PPP link using IPCP address negotiation

```
awplus#configure terminal
awplus(config)#interface ppp0
awplus(config-if)#ip address negotiated
```

See the **ip address negotiated** command for a detailed command description and command examples.

Verifying PPP link configuration

Use the **show running-configuration** and the **show running-configuration interface** commands to verify configuration of a PPP link on an interface.

Use the **show interface (PPP)** command to show the PPP interface status and counters.

Verifying and debugging commands

- **show interface ppp<ppp_index>**

  Displays status and counter statistics for all PPP interfaces configured on the router or access server.

- **debug ppp [interface <ppp-interface-list>][authentication]** and **undebug ppp**

For more information about commands used in this guide, please refer the product's Command Reference.

PPP with authentication

PPP can be authenticated by either:

- "PAP (Password Authentication Protocol)" on page 11, or
- "CHAP (Challenge Handshake Authentication Protocol)" on page 12**,** or
- "EAP (Extensible Authentication Protocol)" on page 13

PAP (Password Authentication Protocol)

PPP defines an extensible LCP that allows negotiation of an authentication protocol for authenticating its peer before allowing Network layer protocols to transmit over the link. RFC 1334 defines two protocols for authentication. PAP is a very basic two-way process. There is no encryption. The username and password are sent in plain text. If it is accepted, the connection is allowed.

The authentication phase of a PPP session is optional. If used, you can authenticate the peer after the LCP establishes the link and choose the authentication protocol. If it is used, authentication takes place before the Network layer protocol configuration phase begins.

The authentication options require that the calling side of the link enter authentication information. This helps to ensure that the user has the permission of the network administrator to make the call. Peer routers exchange authentication messages.

One of the many features of PPP is that it performs Layer 2 authentication in addition to other layers of authentication, encryption, access control, and general security procedures.

Initiating PAP

PAP provides a simple method for a remote node to establish its identity using a two-way handshake. PAP is not interactive. When the ppp authentication pap command is used, the username and password are sent as one LCP data package, rather than the server sending a login prompt and waiting for a response. After PPP completes the link establishment phase, the remote node repeatedly sends a username-password pair across the link until the sending node acknowledges it or terminates the connection.

At the receiving node, the username-password is checked by an authentication server that either allows or denies the connection. An accept or reject message is returned to the requester.

PAP is not a strong authentication protocol. Using PAP, you send passwords across the link in clear text and there is no protection from playback or repeated trial-and-error attacks. The remote node is in control of the frequency and timing of the login attempts. Nonetheless, there are times when using

PAP can be justified. For example, despite its shortcomings, PAP may be used in the following environments:

- A large installed base of client applications that do not support CHAP
- Incompatibilities between different vendor implementations of CHAP
- Situations where a plain text password must be available to simulate a login at the remote host

CHAP (Challenge Handshake Authentication Protocol)

Once authentication is established with PAP, the authentication mechanism performs no further actions. This leaves the network vulnerable to man-in-the-middle attacks. Unlike PAP, which only authenticates once, CHAP conducts periodic challenges to make sure that the remote node still has a valid challenge response. CHAP is more secure than PAP. It involves a three-way exchange of a shared secret.

After the PPP link establishment phase is complete, the local authenticating router sends a challenge message to the remote peer.

The remote peer being authenticated responds with a value calculated using a one-way hash function, which is typically Message Digest 5 (MD5) based on the secret password and challenge message.

The local authenticating router checks the response against its own calculation of the expected hash value. If the values match, the authenticating router acknowledges the authentication. Otherwise, it immediately terminates the connection.

CHAP provides protection against playback attack by using a variable challenge value that is unique and unpredictable. Because the challenge is unique and random, the resulting hash value is also unique and random.

The use of periodically repeated challenges limits the time of exposure to any single attack, and is used to mitigate a current connection from being hijacked by an intermediate device. The local router or a third-party authentication server is in control of the frequency and timing of the challenges.

CHAP authentication process

If an incoming CHAP request requires no authentication, then CHAP progresses to the next stage. If an incoming PPP request requires authentication, then it can be authenticated against the local user database. Successful authentication progresses to the next stage, while an authentication failure will disconnect and drop the incoming PPP request. The PPP interface of the router being authenticated will be configured to provide a secret password to the authenticator. The Authenticator will be configured to compare the received secret password against a user data base.

For example, Router R1 wishes to establish a PPP connection authenticated using CHAP to Router R2:

1. Router R1 (authenticatee) initially negotiates the link connection using LCP with Router R2 and the two Routers agree to use CHAP authentication during the PPP LCP negotiation.
2. Router R2 (authenticator) generates an ID and a random number and sends that plus its username as a CHAP challenge packet to Router R1.
3. Router R1 will then generate a unique MD5 hash number using the Router R2's username, ID, random number and the shared secret password configured on the PPP interface. Router R1 then sends the challenge ID, the hashed value, and its username (Router R1) to Router R2 as its challenge response.
4. Router R2 generates it own hash value using the ID, the shared secret password, and the random number it originally sent to Router R1. Router R2 compares its hash value with the hash value contained in the challenge response sent by Router R1. If the values are the same, Router R2 sends a successful link established acknowledgment response to Router R1.

Note: CHAP and EAP authentication requires a username and password configured in plain text with privilege level 0. PAP authentication may use the default AlliedWare Plus username (login: manager and password: friend).

EAP (Extensible Authentication Protocol)

EAP (Extensible Authentication Protocol) is an authentication framework, not a specific authentication mechanism. EAP provides common functions and negotiation of authentication methods called EAP methods. There are over forty different methods defined in IETF RFCs. EAP is a PPP authentication extension as an alternative to CHAP and PAP authentication. See the table of supported EAP identifier bits and RFC references to look for more information:

EAP IDENTIFIED OPTIONS SUPPORTED IN ALLIEDWARE PLUS (AND THEIR IETF RFC REFERENCES)	
1	Identity (RFC 3748)
2	Notification (RFC 3748)
3	NAK (Response) (RFC 3748)
4	MD5-Challenge (RFC 3748)
19	SRP-SHA1 (Secure Remote Password protocol - Secure Hash Algorithm)

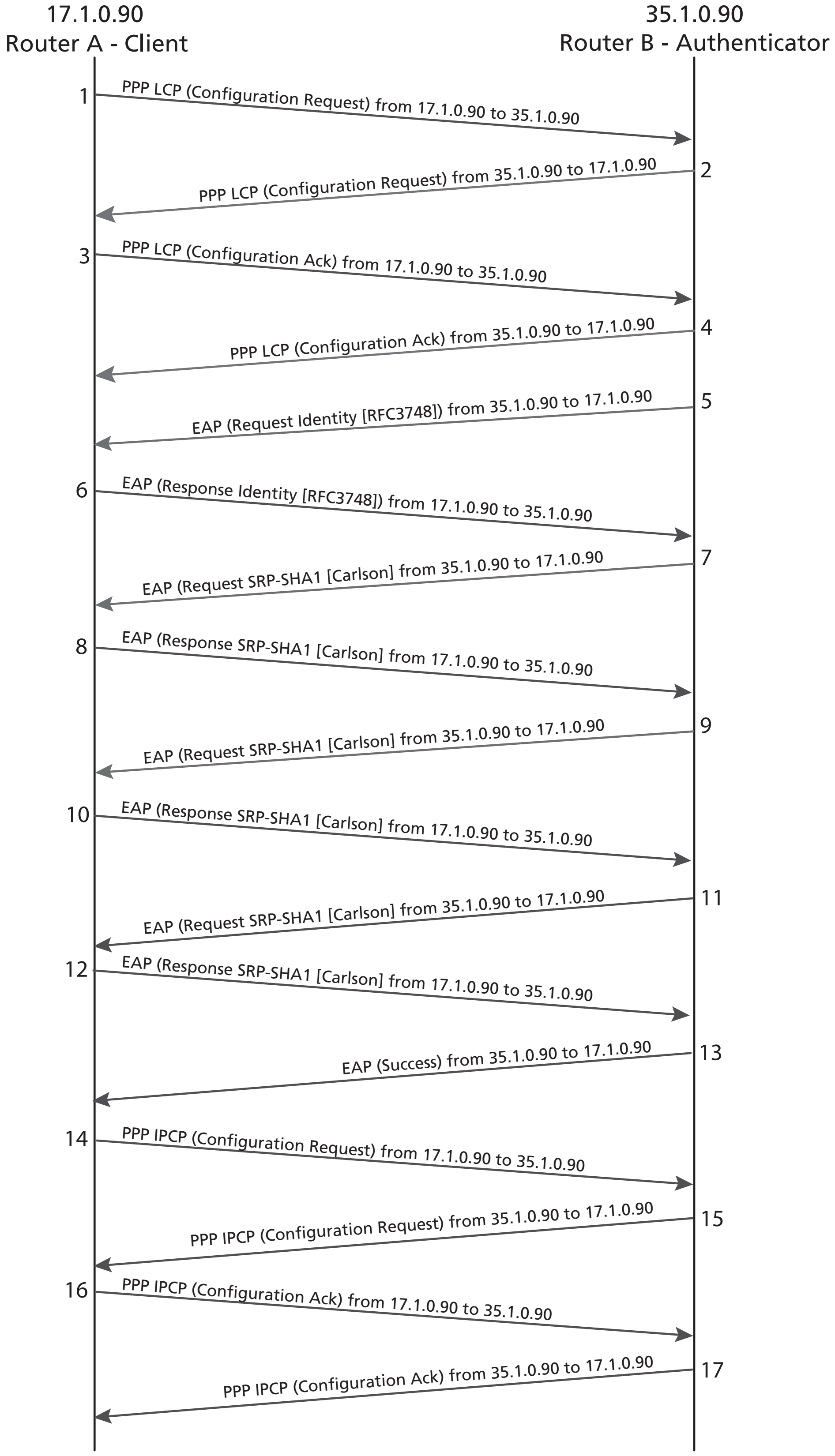
Note: If EAP is configured, then the SRP-SHA1 option is supported by default, but EAP can also automatically fallback to support peers requesting an MD5-Challenge instead.

EAP connection message sequence

The sequence for a successful EAP SRP (Secure Remote Password protocol) -SHA1 (Secure Hash Algorithm) authenticated connection between two routers, Router A and Router B, is listed below. Figure 1 shows the message negotiation from Router A to Router B. Note that Router B would also negotiate LCP, EAP authentication, and IPCP independently and simultaneously, while Router A is negotiating these with Router B.

1. Router A sends a `PPP LCP (Configuration Request)` message to Router B.
2. Router B returns a `PPP LCP (Configuration Request)` message to Router A.
3. Router A sends a `PPP LCP (Configuration Ack)` message to Router A.
4. Router B returns a `PPP LCP (Configuration Ack)` message to Router A
5. Router B sends an `EAP (Request Identity)` message to Router A.
6. Router A returns an `EAP (Response Identity)` message to Router B.
7. Router A sends an `EAP (Request SRP-SHA1)` message to Router B.
8. Router B returns an `EAP (Response SRP-SHA1)` message to Router A.
9. Router A sends an `EAP (Request SRP-SHA1)` message to Router B.
10. Router B returns an `EAP (Response SRP-SHA1)` message to Router A.
11. Router A sends an `EAP (Request SRP-SHA1)` message to Router B.
12. Router B returns an `EAP (Response SRP-SHA1)` message to Router A.
13. Router B sends an `EAP (Success)` message to Router A.
14. Router A sends a `PPP IPCP (Configuration Request)` message to Router B.
15. Router B returns a `PPP IPCP (Configuration Ack)` message to Router A.
16. Router A sends a `PPP IPCP (Configuration Request)` message to Router B.
17. Router B returns a `PPP IPCP (Configuration Ack)` message to Router A.

Figure 1: PPP EAP connection messaging diagram


17.1.0.90
Router A - Client
35.1.0.90
Router B - Authenticator
1 PPP LCP (Configuration Request) from 17.1.0.90 to 35.1.0.90
2 PPP LCP (Configuration Request) from 35.1.0.90 to 17.1.0.90
3 PPP LCP (Configuration Ack) from 17.1.0.90 to 35.1.0.90
4 PPP LCP (Configuration Ack) from 35.1.0.90 to 17.1.0.90
5 EAP (Request Identity [RFC3748]) from 35.1.0.90 to 17.1.0.90
6 EAP (Response Identity [RFC3748]) from 17.1.0.90 to 35.1.0.90
7 EAP (Request SRP-SHA1 [Carlson] from 35.1.0.90 to 17.1.0.90
8 EAP (Response SRP-SHA1 [Carlson] from 17.1.0.90 to 35.1.0.90
9 EAP (Request SRP-SHA1 [Carlson] from 35.1.0.90 to 17.1.0.90
10 EAP (Response SRP-SHA1 [Carlson] from 17.1.0.90 to 35.1.0.90
11 EAP (Request SRP-SHA1 [Carlson] from 35.1.0.90 to 17.1.0.90
12 EAP (Response SRP-SHA1 [Carlson] from 17.1.0.90 to 35.1.0.90
13 EAP (Success) from 35.1.0.90 to 17.1.0.90
14 PPP IPCP (Configuration Request) from 17.1.0.90 to 35.1.0.90
15 PPP IPCP (Configuration Request) from 35.1.0.90 to 17.1.0.90
16 PPP IPCP (Configuration Ack) from 17.1.0.90 to 35.1.0.90
17 PPP IPCP (Configuration Ack) from 35.1.0.90 to 17.1.0.90

PPP authentication configuration

To specify the order in which PPP authentication protocols are requested on the interface, use the **ppp authentication** command (in Interface Configuration mode). Use the **no** form of this command to disable PPP authentication.

After you have enabled CHAP, PAP, or EAP authentication the local router requires the remote device to prove its identity before allowing data traffic to flow. This is done as follows:

- PAP authentication requires the remote device to send a name and password to be checked against a matching entry in the local username database.
- CHAP authentication sends a challenge to the remote device. The remote device must encrypt the challenge value with a shared secret and return the encrypted value and its name to the local router in a response message. It uses the looked-up secret to encrypt the original challenge and verify that the encrypted values match.
- EAP is an authentication protocol for PPP that supports multiple authentication mechanisms. These are negotiated during the authentication phase, instead of during the LCP phase.

You may enable PAP, CHAP, EAP or any combination of these PPP authentication protocols. The order of priority is: EAP, CHAP, PAP. The highest priority authentication protocol that has been configured is requested during link negotiation. If the peer suggests using the second method or simply refuses the first method, the second method is tried. Some remote devices support CHAP only and some PAP only. CHAP is used in preference. If CHAP is rejected, then PAP is used. EAP has the highest priority.

Troubleshooting a PPP authentication configuration

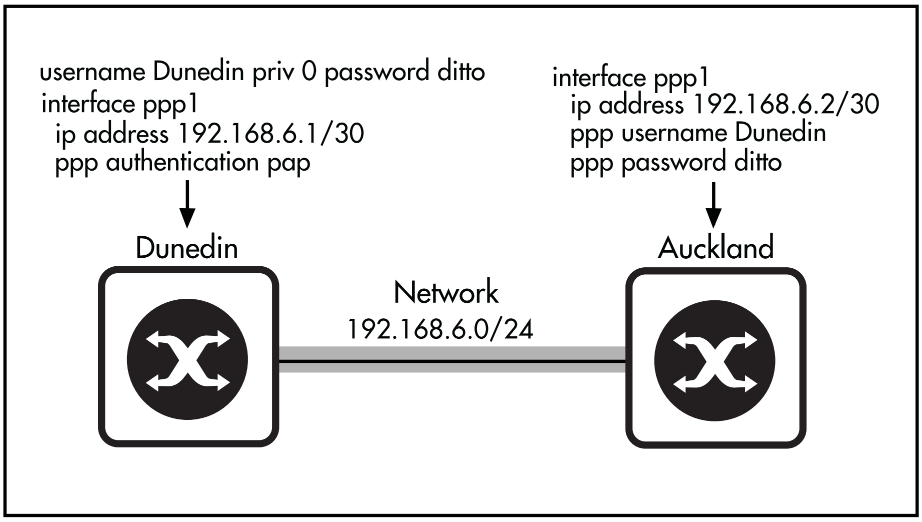
PPP authentication is a feature that needs to be implemented correctly or the security of your serial connection may be compromised. Always verify your configuration with the **show interface** command, in the same way as you would without authentication.

Debugging allows you to confirm your configuration and correct any deficiencies. The command for debugging PPP authentication is **debug ppp**.

PAP authentication configuration

PAP provides a simple method of PPP Authentication for a remote node to establish its identity using a two-way handshake. This is done only on link establishment. PAP does not encrypt the password before transmitting it to the authenticating peer.

Figure 2: Sample PPP PAP authentication network



Output 1: Dunedin to Auckland PPP PAP authentication configuration

```
!
username Dunedin priv 0 password ditto
!
interface ppp1
 ip address 192.168.6.1/30
 ppp authentication pap
!
```

Output 2: Auckland to Dunedin PPP PAP authentication configuration

```
!
interface ppp1
 ip address 192.168.6.2/30
 ppp username Dunedin
 ppp password ditto
!
```

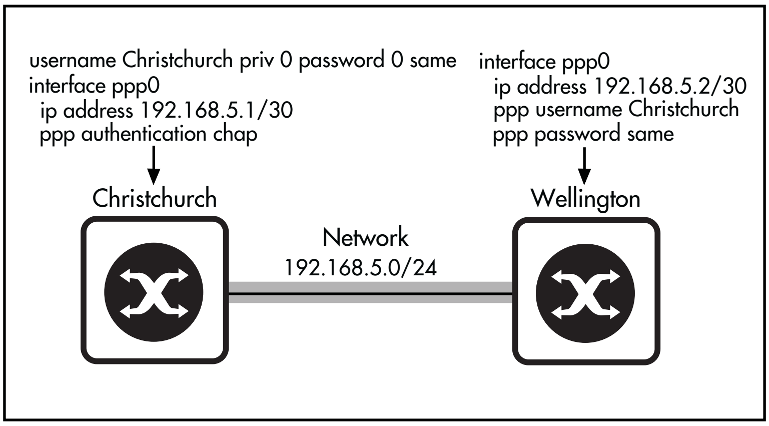
See the **ppp authentication** command for a detailed command description and command examples. See the **username (PPP)** command for a detailed command description and command examples. Note that the PAP password does not need to be stored unencrypted.

CHAP authentication configuration

CHAP periodically verifies the identity of the remote node using a three-way handshake. The authenticator, in this case the Christchurch device, will send the hostname configured by the **ppp hostname** command if configured. Otherwise the authenticator will send the hostname configured by the **hostname** command, or **awplus** if not configured. The passwords must match.

This occurs on initial link establishment and can be repeated any time after the link has been established. Note that CHAP is more secure than PAP since the password is not transmitted across the link in clear text.

Figure 3: Sample PPP CHAP authentication network


username Christchurch priv 0 password 0 same
interface ppp0
ip address 192.168.5.1/30
ppp authentication chap
Christchurch
interface ppp0
ip address 192.168.5.2/30
ppp username Christchurch
ppp password same
Wellington
Network
192.168.5.0/24

Output 3: Christchurch to Wellington PPP CHAP authentication configuration

```
username Christchurch priv 0 password 0 same
! password is stored as plaintext (password 0) in running-config
!
interface ppp0
 ip address 192.168.5.1/30
 ppp authentication chap
!
```

Output 4: Wellington to Christchurch PPP CHAP authentication configuration

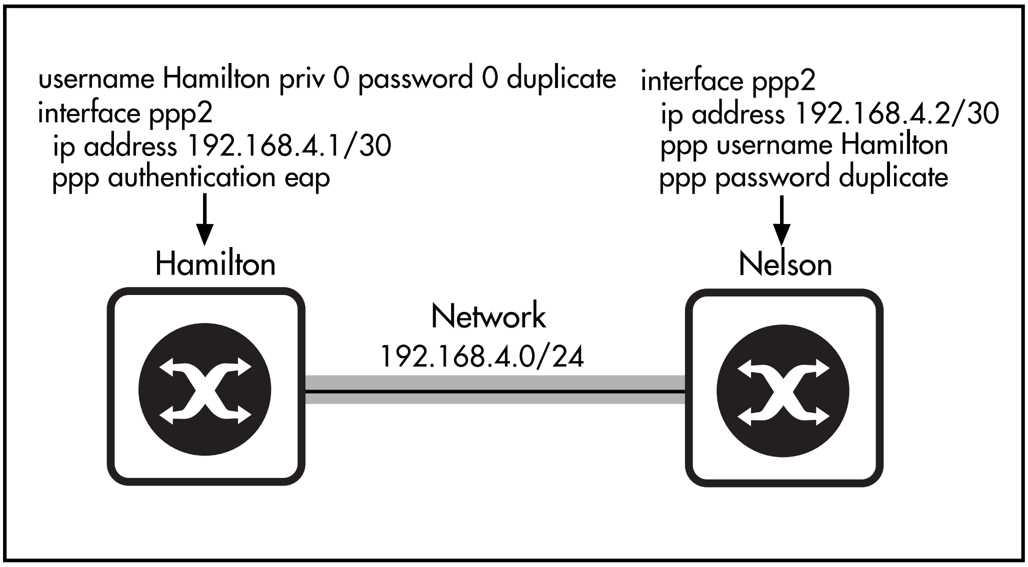
```
!
interface ppp0
 ip address 192.168.5.2/30
 ppp username Christchurch
 ppp password same
!
```

See the **ppp authentication** command for a detailed command description and command examples. See the **username (PPP)** command for a detailed command description and command examples.

EAP authentication configuration

EAP periodically verifies the identity of the remote node using a three-way handshake. The hostname on one router must match the username the other router has configured. The passwords must also match. This occurs on initial link establishment and can be repeated any time after the link has been established. Note that EAP uses a similar mechanism to CHAP but is resistant to dictionary based attacks.

Figure 4: Sample PPP EAP authentication network



Output 5: Hamilton to Nelson PPP EAP authentication configuration

```
username Hamilton priv 0 password 0 duplicate
! password is stored as plaintext (password 0) in running-config
!
interface ppp2
 ip address 192.168.4.1/30
 ppp authentication eap
!
```

Output 6: Nelson to Hamilton PPP EAP authentication configuration

```
!
interface ppp2
 ip address 192.168.4.2/30
 ppp username Hamilton
 ppp password duplicate
!
```

See the **ppp authentication** command for a detailed command description and command examples. See the **username (PPP)** command for a detailed command description and command examples. You will find these in the product's Command Reference.

Point-to-Point Protocol over Ethernet (PPPoE)

PPP over Ethernet, defined in RFC 2516, is a method of transmitting PPP over Ethernet. It provides the ability to connect a network of PPPoE client hosts to a service provider access concentrator over a single bridging access device. A PPPoE link provides a point-to-point connection over a shared medium. An access concentrator may offer multiple services.

PPP over Ethernet enables multiple PPPoE client hosts at a remote site to share the same access device, while providing the access control and accounting functionality of dial-up PPP connections.

PPPoE connectivity stages

PPP over Ethernet has two distinct stages—a discovery stage and a session stage.

In the discovery stage, the PPPoE client discovers all the available access concentrators that offer the required service and then selects one. The client broadcasts a Discovery Initiation packet (PADI), which specifies the name of the required service or indicates that any service is acceptable. If a service name is specified, access concentrators that support the requested service respond with a Discovery Offer packet (PADO) that specifies the access concentrator's unicast Ethernet address. If the client's Initiation packet indicated that any service was acceptable, all access concentrators that have services available respond with a Discovery Offer packet that specifies each access concentrator's unicast Ethernet address.

When the host receives an Offer packet matching its request, it responds by sending a discovery request (PADR) packet specifying the name of the required service to the access concentrator. If it receives more than one valid offer, it responds to the first offer, and ignores the subsequent offers. The access concentrator responds with a Session Confirmation packet (PADS).

When the discovery stage is complete, the host and the selected access concentrator have the information they need to establish the PPPoE connection.

In the session stage, the client host and the access concentrator exchange PPP negotiation packets—such as LCP, authentication, and NCP packets—to establish and maintain the PPP link.

Either the client or the access concentrator can terminate an established PPPoE session any time by sending a Discovery Termination (PADT) packet or a PPP terminate-request.

From software version 5.4.6-1.x onwards, if an AR-series firewall needs to reconnect a failed PPPoE session, the firewall will actively send a PADT packet to the source of the message to terminate that session. This could be necessary, for example, if there was some data loss via the WAN connection, or if the ISP reset the PPP connection. Sending a PADT packet allows PPPoE to reconnect quickly because it does not have to wait for the device at the other end to time out the old session.

PPPoE on the device

The device can be configured as an access concentrator. Remote devices can access services configured on the device. This device can also be configured as a PPPoE client host, creating PPPoE links to services on access concentrators.

We recommend enabling LCP echo keepalive messages, so that PPP can detect a failure of the access concentrator (or the link to it), and attempt to reestablish the connection.

PPPoE uses 8 bytes of the Ethernet frame as overhead. This reduces the maximum size of IP (IPv4 or IPv6) packets that can be transmitted without fragmenting (MTU) from 1500 bytes to 1492 bytes. In order to prevent unnecessary fragmentation of IP packets, the device automatically sets the maximum size of IP packets it transmits over a PPPoE interface to 1492; we recommend also setting end hosts to limit IP packet size to 1492 bytes.

Configuring PPPoE

To configure the device as a PPPoE client, use the procedure below. This procedure can be used to create multiple PPPoE connections via eth interfaces.

Before you configure a PPPoE connection obtain the following information:

- the PPPoE service-name for the connection, or whether to use the default service. This is usually supplied by the service provider.
- which eth interface to use for the PPPoE client connection.
- appropriate PPP negotiation settings. This includes any username, password, and IP or IPv6 address settings.

Note: A PPPoE link cannot be combined with another PPPoE link via Multi-link PPP (ML-PPP).

Table 1: Procedure for configuring the device as a PPPoE client

Step 1. **Create a PPPoE interface**	

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

`awplus(config)#`
`interface eth<eth-id>`
	Enter Interface Configuration mode for the eth interface on which a PPPoE connection will be configured (**interface (PPP)** command).

`awplus(config-if)#`
`encapsulation ppp <ppp-index>`
	Create one or more PPPoE interfaces (**encapsulation ppp** command).

Step 2. **Specify a PPPoE service**	

`awplus(config-if)#`
`interface ppp<ppp-index>`
	Select the PPP interface.

Table 1: Procedure for configuring the device as a PPPoE client (continued)

`awplus(config-if)#`
`ppp service-name <service-name>`
	Optional: Specify a PPPoE service name (**ppp service-name** command). This is the access concentrator service that the PPPoE client will request to connect to. Any access concentrator offering this service will respond. Default: If you do not specify a service-name, the PPP interface will request service-name ANY, and access concentrators will respond by offering their default service.

Step 3. **Configure other settings for the PPP interface**	

`awplus(config-if)(config-if)#`
`keepalive [interval <1-600> attempts <1-10>]`
	LCP echo keepalive request messages are disabled by default. For PPPoE, we recommend enabling them so that PPP can detect a failure of the access concentrator (or the link to it), and attempt to re-establish the connection. The default settings are likely to work well in most networks, but you can modify them if required:
	■ Specify the interval in seconds (1to 600) between LCP Echo keepalive request messages (default: 10).
	■ Specify the number of missing LCP Echo keepalive response messages (1 to 10) before the link is considered to be link down and link renegotiation starts to re-establish the link (default: 3).

`awplus(config-if)#`
`ip address negotiated [<default-ip-address>]`
	Configure dynamic IP addressing for the PPPoE interface—either:
	■ enable the device to obtain an IPv4 address (**ip address negotiated** command), or
	■ enable IPv6 on the device (e.g. **ipv6 address** command, **ipv6 enable** command, or **ipv6 address autoconfig** command), so that it obtains an IPv6 address from the Access Concentrator via PPP address negotiation.

Step 4. **Confirm the PPPoE configuration**	

`awplus(config-if)#`
`end`
	Return to Privileged Exec mode.

`awplus#`
`show interface ppp<ppp_index>`
`awplus#`
`show running-configuration`
`awplus#`
`show running-configuration interface ppp<ppp-index>`
	Display the PPPoE connection settings (**show interface (PPP)** command, **show running-config** command, **show running-config interface** command).

Configuration example

The script extract shown in the output below, shows three PPPoE interfaces over a single eth interface, and enables IPv6 routing on the device. Two of the connections are to specific IPv6 services; the other connects to any default IPv4 service offered by an Access Concentrator.

Output 7: Example configuration for PPPoE

```
!
interface eth1
 encapsulation ppp 5
 encapsulation ppp 6
 encapsulation ppp 7
!
interface ppp5
 ppp service-name ipv6A
 keepalive interval 5
 ipv6 address autoconfig
!
interface ppp6
 ppp service-name dualstack
 keepalive interval 5
 ip address negotiated
 ipv6 address autoconfig
!
interface ppp7
 keepalive interval 5
 ip address negotiated
!
ipv6 forwarding
```

Troubleshooting PPPoE

To enable debugging for a specified PPP interface or for all PPP interfaces on the device, use the **debug ppp** command.

PPPoE relay

Version 5.5.0-1.4 supports PPPoE relay.

PPPoE is a common deployment method for ISPs, allowing them to utilize PPP facilities for identifying and authenticating individual users.

The PPPoE RFC 2516 allows for an intermediate relay device, located between the Host operating as a PPPoE client and an Access Concentrator. This device relays the PPPoE Active Discovery packets and the subsequent PPPoE session packets between the Host and the Access Concentrator as though they are in the same Layer 2 domain.

PPPoE relay tracks state information for multiple Layer 2 PPPoE sessions, and allows multiple PPPoE client connections to be relayed between one or more client LANs and a WAN, allowing access to one or more service provider PPPoE Access Concentrators - whilst at the same time allowing Layer 3 IP traffic routing from the internal LAN(s) to the Internet.

How does PPPoE relay work?

In brief, the process that occurs in PPPoE relay is:

- PPPoE relay listens for PPPoE Active Discovery Initiation (PADI) messages on all configured PPPoE client connected interfaces.
- PADI messages received on client connected interfaces have a relay session ID header added, and are broadcast to all server connected interfaces.
- The PPPoE messages including PPPoE Active Discovery Offer (PADO), PPPoE Active Discovery Request (PADR), and PPPoE Active Discovery Session-confirmation (PADS) are relayed between the client and server for matching session IDs to setup a PPPoE session.
- PPPoE Active Discovery Termination (PADT) messages matching a valid session ID terminates the setup connection.

Configuring PPPoE relay

For example, to create a PPPoE relay configuration instance, and add a client and server interface to the instance, use the following commands:

Step 1. **Create a PPPoE relay instance**	

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

`awplus(config)#`
`pppoe-relay <relay-name>`
	Create a PPPoE relay configuration instance.

Step 2. **Add a PPPoE client connected interface to the configured PPPoE relay**	Add a client to the PPPoE relay interface

`awplus(config-pppoe-relay)#`
`client <client-interface>`	

Step 3. **Add a PPPoE server connected interface to the configured PPPoE relay**	Add a server to the PPPoE relay interface

`awplus(config-pppoe-relay)#`
`server <server-interface>`	

Step 4. **Configure the maximum concurrently PPPoE relay sessions allowed per PPPoE relay instance, the default is 5000.**	(Optional) Configure max sessions

`awplus(config-pppoe-relay)#`
`max-sessions 50`	

Step 5. **Configure the timeout to terminate idle relayed PPPoE sessions that haven't been terminated with a PADT, the default is 600 seconds.**	(Optional) Configure timeout

`awplus(config-pppoe-relay)#`
`timeout 800`	

Configuration example

The following router configuration has two instances of PPPoE relay: Telco1and Telco2.

- Telco1 has a single interface connected to their ISP, eth1. The clients connect to the router via vlan1 and vlan2. The router limits the number of relayed sessions to 10. The timeout on idle relayed PPPoE sessions has been increased from the default of 600 to 750 seconds.
- Telco2 has a simpler configuration, just one client and server interface eth2 and vlan3, respectively.

```
pppoe-relay Telco1
 server eth1
 client vlan1
 client vlan2
 max-sessions 10
 timeout 750
!
pppoe-relay Telco2
 server eth2
 client vlan3
```

Note: Firewall rules do not apply to IP traffic encapsulated with a **relayed** PPPoE session. This is because the relayed PPPoE frames are Layer 2 Ethernet frames, not Layer 3 IP packets. Relayed PPPoE sessions can only be initiated from a client interface, and PPPoE sessions can only be offered via an Access Concentrator interface.

Security of the associated PPPoE client operating on the workstation is the responsibility of the administrator of that device through such things as:

- the use of firewall functionality on the client workstation
- reliance on carrier-grade NAT at the ISP.

IPv6 over PPP

The PPP protocol provides a standard, vendor-independent, method of transporting multiple network layer protocols over a single point-to-point link. It also incorporates a family of Network Control Protocols (NCPs) in order to manage these different network layer protocols.

To encapsulate IPv6 packets in PPP requires a specific IPv6 NCP. RFC 5072 specifies a standard encapsulation method. This RFC defines the following aspects of PPP and IPv6 transmission, which are based on this standard:

- a method of transporting IPv6 packets over PPP links
- an NCP for establishing and configuring IPv6 over PPP
- a method for establishing and configuring IPv6 over PPP

The standard also specifies methods and procedures for managing IPv6 link-local addresses over PPP links.

Once PPP has established the data link using LCP, negotiated any optional facilities, and successfully completed the authentication phase, it enters the network negotiation phase. During this phase it sends Network Control Protocol messages to select and negotiate the required network protocols, such as IPv6. Once this process is complete, IPv6 packets can be sent and received across the link.

In an IPv4/IPv6 dual-stack network, the Link Control Protocol (LCP) allows for the transport of both IPv4 and IPv6 traffic at the same time over a single PPP session. PPP negotiates and then runs the two NCP's independently in parallel—IPCP for IPv4 and IP6CP for IPv6. The protocol field in the PPP header distinguishes IPv4 traffic (0x0021) from IPv6 traffic (0x0057).

In contrast to IPCP, which provides other configuration information for IPv4 (such as DNS information), IP6CP negotiates only an IPv6 interface identifier (IPv6 link-local address). To dynamically configure an IPv6 global unicast address, IPv6 uses either stateless address autoconfiguration (SLAAC) or DHCPv6. Router solicitations and router advertisements are still useful on PPP links for router discovery, as are other functions of IPv6 neighbor discovery.

For IPv6, the Service Provider Access Concentrator (AC) can assign an IPv6 global address to the remote peer via SLAAC or DHCPv6. The AC end of the PPP link typically has only a link-local IPv6 address. Whichever method the AC uses to assign an IPv6 global address to the PPP client, it can and should be configured to generate router advertisement messages. These messages can be used by the PPP client to populate its default routers list (RFC 4861) and to optionally construct IPv6 SLAAC addresses on the WAN interface.

Note: The IPv6 global unicast address delegated to the device via the PPP link does not have to be configured on the PPP client interface. It is also common for the PPP link to only contain a link-local address only, with the IPv6 global unicast addressing information associated with another interface.

Currently the router supports the use of link-local addressing on PPP interfaces, as well as static addressing and SLAAC (Stateless Address Auto Configuration), or DHCPv6 prefix delegation.

DHCPv6 via PPPoE WAN

From release 5.4.8-1.x onwards, the router PPPoE WAN interface can be optionally configured as a DHCPv6 prefix delegation client.

When configured, PPP IPv6 Control Protocol negotiation occurs, and the IPv6 PPP link is established with link-local addressing only.

The router then requests allocation of a globally scoped IPv6 prefix from an ISP router to the router DHCPv6 IAPD client via the established PPP link.

The dynamically learned IPv6 prefix is then stored within a named DHCPv6 prefix delegation pool, and is used to configure internal LAN interfaces with a globally scoped IPv6 address.

Configuration example

The following example shows how to configure the router PPP interface as a DHCPv6 prefix delegation client.

The DHCPv6 prefix is stored in a named prefix delegation pool, and the internal VLAN IPv6 global address is configured based on concatenation of EUI-64 and the learned prefix. The IPv6 DNS server address is also learned dynamically via DHCPv6 from the service provider.

The IPv6 default route via the PPP link is dynamically created based on the RA received from the service provider via the PPP link. RA suppression should also be disabled on the internal VLAN interface to enable any clients attached to the LAN to use SLAAC to configure their own global IPv6 addressing, and default route.

Additionally, IPv6 firewall rules are configured to trust and allow DHCPv6 and ICMPv6 traffic from the link-local subnet from the service provider peer to be received via the PPP WAN. This allows any traffic originating from the PPP WAN and also from the internal VLAN access to the IPv6 Internet.

Output 8: Example configuration for the AR-Series Firewall PPP interface as a DHCPv6 prefix delegation client:

```
!
hostname AW+
!
zone lan
 network lan
  ipv6 subnet ::/0 interface vlan1
!
zone public
 network internet
  ipv6 subnet ::/0 interface ppp0
 network link_local
  ipv6 subnet fe80::/64
 network wan
  host ppp
   ipv6 address dynamic interface ppp0
!
application dhcpv6
 protocol udp
 sport 546 to 547
 dport 546 to 547
!
application icmpv6
 protocol ipv6-icmp
!
```

Example configuration for the AR-Series Firewall PPP interface as a DHCPv6 prefix delegation client (cont’d):

```
firewall
 rule 10 permit any from public.wan.ppp to public.internet
 rule 20 permit any from lan to public.internet
 rule 30 permit dhcpv6 from public.link_local to public.wan.ppp
 rule 40 permit icmpv6 from public.link_local to public.wan.ppp
 protect
!
no ip domain-lookup
!
interface eth1
 encapsulation ppp 0
 no ipv6 nd accept-ra-pinfo
!
interface vlan1
 ipv6 enable
 no ipv6 nd suppress-ra
 no ipv6 nd accept-ra-pinfo
 ipv6 address pool1 ::/64 eui64
!
interface ppp0
 ppp service-name any
 ppp username <username>
 ppp password <password>
 ipv6 enable
 ipv6 tcp adjust-mss pmtu
 ipv6 dhcp client pd pool1
!
ipv6 forwarding
!
ip dns forwarding
ip dns forwarding cache size 100 timeout 500
!
```

Some useful diagnostics show commands are as follows:

Output 9: Example show commands:

```
awplus#show ipv6 dhcp interface

ppp0 is in client (Prefix-Delegation) mode
 Prefix name pool1
        prefix 2001:db8:1200::/48
        preferred lifetime 600, valid lifetime 1800
        starts  at 30 Apr 2018 02:28:50
        expires at 30 Apr 2018 02:58:50
```

```
awplus#show ip name-server

Currently learned name-servers:
2001:db8:4860::8888 dynamic (ppp0)
```

Example show commands: (cont'd)

```
awplus#show ipv6 interface brief

  * = Autoconfigured Address
Interface     IPv6-Address                          State        Status      Protocol
eth1          unassigned                            N/A          admin up    running
eth2          unassigned                            N/A          admin up    down
lo            unassigned                            N/A          admin up    running
vlan1         2001:db8:1200:0:21a:ebff:fe93:7acd/64 preferred    admin up    running
              fe80::21a:ebff:fe93:7acd/64            preferred
ppp0          fe80::21a:ebff:fe93:7acd/10            preferred    admin up    running
```

```
awplus#show ipv6 route

IPv6 Routing Table
Codes: C - connected, S - static, R - RIP, O - OSPF, B - BGP, D - DHCP
       IA - OSPF inter area E1 - OSPF ext. type 1, E2 - OSPF ext. type 2
Timers: Uptime

C      ::/0 via fe80::32e4:dbff:fe5f:e402, ppp0, 00:02:22
S      2001:db8:1200::/48 [1/0] via ::, Null, 00:02:23
C      2001:db8:1200::/64 via ::, vlan1, 2d21h45m
C      fe80::/10 via ::, ppp0, 00:02:26
C      fe80::/64 via ::, vlan1, 2d21h45m
```

```
awplus#show interface ppp0

Interface ppp0
  Link is UP, administrative state is UP
  Hardware is PPP
  IPv6 address fe80::21a:ebff:fe93:7acd/10
  index 30000 metric 1 mtu 1492
  IPv6 mss 1432
  <UP,POINT-TO-POINT,RUNNING,NOARP,MULTICAST>
  VRF Binding: Not bound
  PPP is running over interface eth1
  LCP Opened IPV6CP Opened
  MRU(bytes): Local config 1492, Local negotiated 1492, Peer negotiated 1492
  Magic number: Local config ON, Local negotiated ON, Peer negotiated ON
  Authentication: Local config None, Local neg None, Peer neg CHAP
  IPv4 addresses: Local config 0.0.0.0
  IPv6 Id Local config: 021a:ebff:fe93:7acd
  IPv6 Id Neg: Local 021a:ebff:fe93:7acd, Peer 32e4:dbff:fe5f:e402
  PPPoE service name is any
  SNMP link-status traps: Disabled
  Router Advertisement is disabled
  Router Advertisement default routes are accepted
  Router Advertisement prefix info is accepted
    input packets 22, bytes 1512, dropped 0, multicast packets 0
    output packets 5, bytes 319, multicast packets 0, broadcast packets 0
    input average rate : 30 seconds 0 bps, 5 minutes 0 bps
    output average rate: 30 seconds 0 bps, 5 minutes 0 bps
    input peak rate 243 bps at 2018/04/30 02:28:56
    output peak rate 108 bps at 2018/04/30 02:28:56
  Time since last state change: 0 days 00:02:34
```

Example show commands: (cont’d)

```
awplus#show firewall rule

[* = Rule is not valid - see "show firewall rule config-check"]
 ID     Action  App               From             To               Hits
--------------------------------------------------------------------------------
 10     permit  any               public.wan.ppp   public.internet  24
 20     permit  any               lan              public.internet  6
 30     permit  dhcpv6            public.link_local
                                                   public.wan.ppp   23
 40     permit  icmpv6            public.link_local
                                                   public.wan.ppp   1
```

```
awplus#show firewall rule config-check
All rules are valid
```

PPP IP Borrow

PPP IP Borrow is used to process IP packets on an PPP interface without explicitly assigning an IP address. This action can be performed by borrowing an IP address from another interface. The PPP interface which borrows the IP address is called the unnumbered interface. This feature is especially useful to save the scarce IPv4 addresses.

You can use the **ip unnumbered** command to borrow primary IPv4 address from specified interfaces like VLAN, loopback, Ethernet and bridge. The IP address will be borrowed from the specified interface regardless of the operational state of the interface, for example, it doesn't matter if a VLAN or Ethernet interface is up or down, the IP address will be used. A packet originating from the AlliedWare Plus device's unnumbered PPP interface has borrowed the IP address as its source IP address. PPP interface advertises the borrowed IP address, while locally using 0.0.0.0 on its interface, cannot be used as IP interface.

Dynamic routing protocols such as OSPF and BGP are allowed to operate over unnumbered PPP interface. You can use the **no peer neighbor-route** command to suppress the connected host route (with subnet mask /32) that AlliedWare Plus normally creates for the PPP interface.

PPP Dial on Demand

PPP Dial on Demand is used where a network connection is established only as required. Using PPP with this technique, a direct connection between two nodes can be established based on certain criteria, generally when the data needs to be sent.

On AlliedWare Plus devices, PPP links can be established over L2TP and PPPoE and can transport both IPv4 and IPv6 protocols.

An idle timer will disconnect the connection after certain seconds determined by the configured device. This would be reset upon either ingress or egress user traffic. Non-user traffic such as Link Control Protocol (LCP) keepalives and Network Control Protocol (NCP) negotiation packets do not reset the idle timer. You can use the **ppp timeout idle** command to enable the PPP Dial on Demand feature by specifying an idle time when a ppp connection is disconnected.

MTU and MSS

The Maximum Transmission Unit (MTU) is the maximum number of bytes per packet that may be transmitted by the interface. If a single packet exceeds the MTU, the packet is divided into smaller packets before transmission.

For a TCP packet, the packet size is the header size plus the MSS (Maximum Segment Size), where the header size is the size of the packet header and the MSS or TCP MSS so the segment will adjust its size before reaching the data link layer.

Before we look at TCP MSS, it helps to understand the build of the “unit” that’s being sent over the Internet is the largest amount of TCP data (in bytes) that the device can transmit or receive in one single data packet. To avoid fragmentation related issues, the MSS must be less than the MTU by at least an amount equal to the packet header size.

The MTU size for an interface is set manually with the **mtu** command. See the **mtu** command for details about setting the MTU manually.

If MSS clamping is configured on an interface, then the MSS value in TCP SYN packets egressing the interface will be examined. If the MSS value in a given TCP SYN is too large for the interface, then the MSS is reduced to a suitable value. The definition of “too large” depends on the way in which MSS clamping has been configured on the interface.

- MSS clamping has been configured with a specific MSS value. TCP SYNs egressing the interface have an MSS value greater than the configured MSS value, then their MSS is reduced to the configured value.
- MSS clamping has been configured to calculate the maximum MSS value from the interface's MTU. If the MSS value in an egressing TCP SYN is larger than the value of MTU-<header size>, then the packet's MSS is reduced to the value of MTU-<header size>.

The MSS value in the TCP SYN is rewritten, and the header checksum is recalculated and rewritten into the packet which is then sent on its way.

Configuring MSS clamping

MSS clamping is configured in Interface Configuration mode on the interface out which the packets will be egressing.

You can configure MSS clamping for IPv4 by using the **ip tcp adjust-mss** command. You can also configure MSS clamping for IPv6 by using the **ipv6 tcp adjust-mss** command.

If a particular value is specified in the command, then the MSS will be set to that value. For PPPoE links, the MSS value should be set based on the interface MTU, using the following formula:

- for IPv4: MSS = MTU-40
- for IPv6: MSS = MTU-60

In the case of a PPPoE connection, the optimum value for the MSS parameter is 1452 bytes. This value plus the 20-byte IP header, the 20-byte TCP header, and the 8-byte PPPoE header add up to a 1500-byte packet that matches the MTU size for the Ethernet link.

For example, it is recommended that you use the following commands and values, as part of the PPPoE interface configuration for IPv4 via PPP:

```
awplus#configure terminal
awplus(config)#interface ppp0
awplus(config-if)#ip tcp adjust-mss 1452
awplus(config-if)#mtu 1492
```

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