

# Junos® OS

---

## Layer 2 Network Access Protocols User Guide

Published  
2023-12-11

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

*Junos® OS Layer 2 Network Access Protocols User Guide*  
Copyright © 2023 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

## YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

About This Guide | vi

1

## Overview

Network Access Overview | 2

2

## Configuring PPP and L2TP

Point-to-Point Protocol (PPP) | 4

PPP Configuration | 5

Configure PPP Encapsulation | 5

Example: PPP Encapsulation | 6

Configure LCP | 7

Configure NCP | 8

PPP Clear Loop Detected Timer for LCP | 9

Verify the LCP and NCP Configuration | 9

PPP Profiles | 10

Access Profiles | 10

Client-Specific Profile | 11

Group Profiles | 13

Compress PPP Fields | 15

Address and Control Field Compression | 15

Protocol Field Compression | 16

Monitor PPP Field Compression | 17

Configure PPP on ACX Series Routers | 18

Prepare Interfaces | 19

Configure PPP Encapsulation | 19

Monitor a PPP Session | 20

Monitor the PPP Process | 20

Layer 2 Tunneling Protocol (L2TP) | 21

Minimum L2TP Configuration | 22

L2TP Profiles | 23

Access Profiles | 24

Group Profile | 29

Reference the Group Profile from the L2TP Profile | 30

Example: L2TP Multilink PPP Support on Shared Interfaces | 30

Example: PPP MP for L2TP | 31

Configure L2TP Authentication | 32

Configure the CHAP Secret for an L2TP Profile | 32

Example: Configuring L2TP PPP CHAP | 33

Configure the PAP Password for an L2TP Profile | 34

Example: Configure PAP for an L2TP Profile | 34

Example: Configure L2TP | 35

Configure L2TP for M7i and M10i Routers | 37

Address Pool for L2TP Network Server IP Address Allocation | 39

IKE Access Profiles | 41

### 3

## Configuring Authentication for PPP and L2TP

PPP Challenge Handshake Authentication Protocol | 44

PPP Challenge Handshake Authentication Protocol | 44

Configuring the PPP Challenge Handshake Authentication Protocol | 44

Displaying the Configured PPP Challenge Handshake Authentication Protocol | 47

Example: Configuring PPP CHAP | 48

Example: Configure CHAP Authentication with RADIUS | 49

Configuration | 50

PPP Password Authentication Protocol | 53

Understanding PAP | 53

Configure PAP on a Physical Interface | 54

Configure PAP on a Logical Interface | 55

RADIUS Authentication for L2TP | 57

Configure RADIUS Authentication for L2TP | 57

Configure RADIUS Authentication for an L2TP Client and Profile | 59

RADIUS Local Loopback Interface Attribute for L2TP | 60

Example: Configure RADIUS Authentication for L2TP | 61

| Configuration | 61

Example: Configure RADIUS Authentication for an L2TP Profile | 62

| Configuration | 62

Configure the RADIUS Disconnect Server for L2TP | 63

Configure RADIUS Accounting Order for L2TP | 64

Example: Configure RADIUS-Based Subscriber Authentication and Accounting | 65

| Configuration | 65

RADIUS Attributes for L2TP | 68

**Subscriber Session Timeout Options | 74**

## **Configuration Statements and Operational Commands**

**Junos CLI Reference Overview | 77**

# About This Guide

Use this guide to configure common Layer 2 protocols.

## RELATED DOCUMENTATION

| [Junos OS Portable Libraries](#)

# 1

CHAPTER

## Overview

---

[Network Access Overview](#) | 2

---

# Network Access Overview

Junos OS enables you to configure network access features for the device at the [edit access] hierarchy level. Network access features include Layer 2 Tunneling Protocol (*L2TP*), Point-to-Point Protocol (*PPP*), and *Subscriber Access* configuration. PPP is an encapsulation protocol for transporting IP traffic across point-to-point links. L2TP allows PPP to be tunneled within a network.

For information about configuring Subscriber Access, see [Broadband Subscriber Sessions User Guide](#). For information about multilink PPP (MLPPP), see [Link and Multilink Services Interfaces User Guide for Routing Devices](#).



# 2

CHAPTER

## Configuring PPP and L2TP

---

Point-to-Point Protocol (PPP) | 4

Layer 2 Tunneling Protocol (L2TP) | 21

Address Pool for L2TP Network Server IP Address Allocation | 39

IKE Access Profiles | 41

---

# Point-to-Point Protocol (PPP)

## IN THIS SECTION

- [PPP Configuration | 5](#)
- [PPP Profiles | 10](#)
- [Compress PPP Fields | 15](#)
- [Configure PPP on ACX Series Routers | 18](#)
- [Monitor a PPP Session | 20](#)
- [Monitor the PPP Process | 20](#)

Point-to-Point Protocol (PPP) is a Layer 2 communications protocol. PPP encapsulates multiprotocol data over point-to-point links. PPP encapsulation is the default encapsulation type for physical interfaces.

To configure PPP for subscriber access, see *PPP Subscriber Access Networks Overview*.

## Benefits of PPP

- Flexible
- Built-in testing of the link to reduce packet loss
- Can encapsulate multiple protocols simultaneously on the same link

## Limitations of PPP

- IP class of service (CoS) is not supported on PPP interfaces. All the traffic is sent to the best effort queue (queue 0) and CoS code points are not processed.
- Fixed classifiers are not supported.
- The MPLS family is not supported on logical interfaces if you configured PPP encapsulation on the interface.
- The circuit cross-connect (CCC) version of PPP (configured with the `ppp-ccc` option) and the translational cross-connect (TCC) version of PPP (`ppp-tcc` option) are not supported for configuration with the `encapsulation` statement.

## Supported PPP Interface Standards

Junos OS substantially supports the following RFCs, which define standards for PPP interfaces.

- RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*
- RFC 1334, *PPP Authentication Protocols*
- RFC 1661, *The Point-to-Point Protocol (PPP)*

## PPP Configuration

### IN THIS SECTION

- [Configure PPP Encapsulation | 5](#)
- [Example: PPP Encapsulation | 6](#)
- [Configure LCP | 7](#)
- [Configure NCP | 8](#)
- [PPP Clear Loop Detected Timer for LCP | 9](#)
- [Verify the LCP and NCP Configuration | 9](#)

## Configure PPP Encapsulation

To configure PPP encapsulation on a physical interface:

1. In configuration mode, go to the `[edit interfaces interface-name]` hierarchy level.

```
[edit]
user@device# edit interfaces interface-name
```

2. To enable PPP encapsulation, include the `encapsulation ppp` statement.

```
[edit interfaces interface-name]
user@device# set encapsulation ppp
```

3. (Optional) Configure PPP-specific interface properties by including the `ppp-options` statement. Details of these options are explained in later sections.

```
[edit interfaces interface-name]
user@device# set ppp-options
```

You can use the following operational mode commands to view the PPP configuration and statistical details:

- `show ppp address-pool` displays PPP address pool information.
- `show ppp interface` displays PPP session information for an interface.
- `show ppp statistics` displays PPP session statistics.
- `show ppp summary` displays summary information about PPP-configured interfaces.
- `show interfaces e1-fpc/pic/port`, `show interfaces t1-fpc/pic/port`, and `show interfaces ds-fpc/pic/port` display the PPP settings of a specific E1, T1, or DS interface, respectively.

## SEE ALSO

| [encapsulation](#)

## Example: PPP Encapsulation

Use this example to configure PPP encapsulation on a SONET/SDH interface. The second and third family statements allow IS-IS and MPLS to run on the interface.

```
[edit interfaces]
so-7/0/0 {
  encapsulation ppp;
  unit 0 {
    point-to-point;
    family inet {
      address 192.168.1.113/32 {
        destination 192.168.1.114;
      }
    }
    family iso;
    family mpls;
```

```
}
}
```

## Configure LCP

PPP uses the Link Control Protocol (LCP) to establish and test a link before transmitting data. LCP can negotiate optional configurations for the link with the other device such as the method used to authenticate the link. To establish and test the link, LCP sends different types of packets to the peer device.

1. In configuration mode, go to the correct hierarchy.

```
[edit]
user@device# edit interfaces interface-name unit number ppp-options
```

2. The first packet LCP sends is the Configure-Request packet, which is a request to configure the link. The device sends LCP Configure-Requests until it receives a response or reaches the specified maximum number. By default, the device sends a maximum of 254 Configure-Request packets. To configure a different maximum number of LCP Configure-Request packets:

```
[edit interfaces interface-name unit number ppp-options]
user@device# set lcp-max-conf-req number
```

The *number* range is from 0 to 65,535. If you configure the maximum to be 0, there is no limit and the device sends LCP Configure-Requests indefinitely.

3. LCP uses a restart timer to protect against packet loss. The timer starts when LCP sends a packet. When the restart timer expires, the device resends the packet. You can configure the LCP restart timer on interfaces with PPP, PPP TCC, PPP over Ethernet, PPP over ATM, and PPP over Frame Relay encapsulations. By default, the restart time expires after 3 seconds. To change the restart timer expiration time, include the `lcp-restart-timer` statement and specify the number of milliseconds.

```
[edit interfaces interface-name unit number ppp-options]
user@device# set lcp-restart-timer milliseconds
```

You can also configure this statement at the `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number ppp-options]` hierarchy level.

## SEE ALSO

| [\*lcp-max-conf-req\*](#)

| *lcp-restart-timer*

## Configure NCP

After LCP has established a link, PPP uses the Network Control Protocol (NCP) to configure Layer 3 protocols. PPP can use multiple Layer 3 protocols simultaneously. To configure NCP:

1. In configuration mode, go to the correct hierarchy.

```
[edit]
user@device# edit interfaces interface-name unit number ppp-options
```

2. The first packet NCP sends is the Configure-Request packet, which is a request to configure the link. The device sends NCP Configure-Request packets until it receives a response or reaches the specified maximum number. By default, the device sends a maximum of 254 Configure-Request packets. To configure a different maximum number of LCP Configure-Request packets:

```
[edit interfaces interface-name unit number ppp-options]
user@device# set ncp-max-conf-req number
```

The *number* range is from 0 to 65,535. If you configure the maximum to be 0, there is no limit and the device sends LCP Configure-Requests indefinitely.

3. NCP uses a restart timer to protect against packet loss. The timer starts when NCP sends a packet. When the restart timer expires, the device resends the packet. You can configure the NCP restart timer on interfaces with PPP and PPP TCC encapsulations and on multilink PPP bundle interfaces. By default, the restart time expires after 3 seconds. To change the restart timer expiration time, include the `ncp-restart-timer` statement and specify the number of milliseconds.

```
[edit interfaces interface-name unit number ppp-options]
user@device# set ncp-restart-timer milliseconds
```

You can also configure this statement at the `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number ppp-options]` hierarchy level.

## SEE ALSO

| *ncp-max-conf-req*

| *ncp-restart-timer*

## PPP Clear Loop Detected Timer for LCP

When a PPP session detects a loop, the operating system sets the loop detected flag. If the protocol doesn't clear the flag after it clears the loopback, the clear loop detected timer clears the flag after the specified time has elapsed.

To configure the clear loop detected timer for the LCP component of a PPP session, include the `loopback-clear-timer` statement and specify the number of seconds.

```
[edit interfaces interface-name unit logical-unit-number ppp-options]
user@device# set loopback-clear-timer seconds
```

You can also include this statement at the [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* ppp-options] hierarchy level.

To monitor the configuration, issue the `show interfaces interface-name extensive` command.

## Verify the LCP and NCP Configuration

To monitor the LCP and NCP configuration, issue the `show interfaces interface-name` command. The operating system displays the configured options in the PPP parameters field for the physical interface.

```
user@host> run show interfaces t1-0/0/0:1:1.0 detail
Logical interface t1-0/0/0:1:1.0 (Index 67) (SNMP ifIndex 40)
(Generation 156)
Flags: Hardware-Down Device-Down Point-To-Point SNMP-Traps 0x4000
Encapsulation: PPP
PPP parameters:
  LCP restart timer: 2000 msec
  NCP restart timer: 2000 msec
Protocol inet, MTU: 1500, Generation: 163, Route table: 0
Flags: Protocol-Down
Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
  Destination: 198.51.100/24, Local: 198.51.100.2, Broadcast: 198.51.100.255,
```

## PPP Profiles

### IN THIS SECTION

- [Access Profiles | 10](#)
- [Client-Specific Profile | 11](#)
- [Group Profiles | 13](#)

When multiple types of profiles are configured, the operating system only implements one configuration. The operating system prioritizes them as follows, where `[edit access profile profile-name]` overrides all other profile configurations:

1. `[edit access profile profile-name]`
2. `[edit access group-profile profile-name]`
3. `[edit access profile profile-name user-group-profile profile-name]`

**NOTE:** When you configure a profile, you can configure either L2TP or PPP parameters, but not both at the same time.

### Access Profiles

To validate PPP connections and session requests, set up access profiles by configuring the profile statement at the `[edit access]` hierarchy level. You can configure multiple profiles. You can also configure multiple clients for each profile.

1. To configure the access profile, include the profile statement at the `[edit access]` hierarchy level and assign a name to the profile.

```
[edit access]
user@device# set profile profile-name
```

2. You can configure the order in which the operating system tries different methods to authenticate peers. For each access attempt, the software tries the authentication methods in order, from first to



last. Configure the authentication order using the `authentication-order` statement. If you do not include the `authentication-order` statement, the operating system verifies clients using password authentication.

```
[edit access profile profile-name]
user@device# set authentication-order [ authentication-methods ]
```

In *authentication-methods*, specify one or more of the following in the preferred order, from first tried to last tried:

- `radius`—Verify the client using RADIUS authentication services.
- `password`—Verify the client using the information configured at the `[edit access profile profile-name client client-name]` hierarchy level.

## SEE ALSO

| [profile](#)

## Client-Specific Profile

To define PPP properties for a client-specific access profile, include one or more of the following statements at the `[edit access profile profile-name client client-name ppp]` hierarchy level.

**NOTE:** The properties defined in the profile take precedence over the values defined in the group profile.

```
[edit access profile profile-name]
client client-name {
  chap-secret chap-secret;
  group-profile profile-name;
  pap-password pap-password;
  ppp {
    cell-overhead;
    encapsulation-overhead bytes;
    framed-ip-address;
    framed-pool framed-pool;
    idle-timeout seconds;
    interface-id interface-id;
    keepalive seconds;
    primary-dns primary-dns;
```

```

        primary-wins primary-wins;
        secondary-dns secondary-dns;
        secondary-wins secondary-wins;
    }
}

```

The `cell-overhead` statement configures the session to use ATM-aware egress shaping on the IQ2 PIC.

*bytes* (in the `encapsulation-overhead` statement) configures the number of bytes used as overhead for class-of-service calculations.

*ip-address* (in the `framed-ip-address` statement) is the IPv4 prefix.

*pool-id* (in the `framed-pool` statement) is a configured address pool.

*seconds* (in the `idle-timeout` statement) is the number of seconds a user can remain idle before the session is terminated. By default, idle timeout is set to 0. You can configure this to be a value in the range from 0 through 4,294,967,295.

*interface-id* (in the `interface-id` statement) is the identifier for the interface representing an L2TP session configured at the [edit interfaces *interface-name* unit *local-unit-number* dial-options] hierarchy level.

`keepalive seconds` is the time period that must elapse before the Junos OS checks the status of the PPP session by sending an echo request to the peer. For each session, Junos OS sends a maximum of ten keepalives at 10-second intervals and the session is closed if there is no response. By default, the time to send a *keepalive* messages is set to 10 seconds. You can configure this to be a value in the range from 0 through 32,767 seconds.

`keepalive-retries number-of-retries` is the number of retry attempts for checking the keepalive status of a Point-to-Point (PPP) protocol session. Configuring a lower number of retries helps reduce the detection time for PPP client session failures or timeouts if you have configured a *keepalive seconds* value. By default, the number of retries is set to 10 times. You can configure this to be a value in the range from 3 through 32,767 times.

*primary-dns* (in the `primary-dns` statement) is an IPv4 address.

*secondary-dns* (in the `secondary-dns` statement) is an IPv4 address.

*primary-wins* (in the `primary-wins` statement) is an IPv4 address.

*secondary-wins* (in the `secondary-wins` statement) is an IPv4 address.

When you configure PPP properties for a profile, you typically configure the `chap-secret` statement or `pap-password` statement.

## Group Profiles

### IN THIS SECTION

- [Configure PPP for a Group Profile | 13](#)
- [Apply a PPP Group Profile to a Tunnel | 14](#)

If you need to apply PPP to multiple devices, you might want to configure group profile to define the PPP attributes. Any client referencing the configured group profile inherits all the group profile attributes. This makes it easier to apply PPP on a larger scale.

### Configure PPP for a Group Profile

To configure the PPP attributes for a group profile, include the following statements at the [edit access group-profile *profile-name* ppp] hierarchy level:

```
[edit access group-profile profile-name ppp]
cell-overhead;
encapsulation-overhead bytes;
framed-pool pool-id;
idle-timeout seconds;
interface-id interface-id;
keepalive seconds;
ppp-options {
    aaa-options aaa-options-name;
    chap;
    ignore-magic-number-mismatch;
    initiate-ncp (ip | ipv6 | dual-stack-passive)
    ipcp-suggest-dns-option;
    mru;
    mtu;
    pap;
    peer-ip-address-optional;
}
primary-dns primary-dns;
primary-wins primary-wins;
secondary-dns secondary-dns;
secondary-wins secondary-wins;
```

The `cell-overhead` statement configures the session to use Asynchronous Transfer Mode (*ATM*)-aware egress shaping on the IQ2 PIC.

*bytes* (in the `encapsulation-overhead` statement) configures the number of bytes used as overhead for class-of-service calculations.

*pool-id* (in the `framed-pool` statement) is the name assigned to the address pool.

*seconds* (in the `idle-timeout` statement) is the number of seconds a user can remain idle before the session is terminated. By default, idle timeout is set to 0. You can configure this to be a value in the range from 0 through 4,294,967,295.

*interface-id* (in the `interface-id` statement) is the identifier for the interface representing an L2TP session configured at the `[edit interfaces interface-name unit local-unit-number dial-options]` hierarchy level.

*seconds* (in the `keepalive` statement) is the time period that must elapse before the Junos OS checks the status of the PPP session by sending an echo request to the peer. For each session, Junos OS sends out three keepalives at 10-second intervals and the session is close if there is no response. By default, the time to send a keepalive message is set to 10 seconds. You configure this to be a value in the range from 0 through 32,767.

*primary-dns* (in the `primary-dns` statement) is an IP version 4 (IPv4) address.

*secondary-dns* (in the `secondary-dns` statement) is an IPv4 address.

*primary-wins* (in the `primary-wins` statement) is an IPv4 address.

*secondary-wins* (in the `secondary-wins` statement) is an IPv4 address.

## Apply a PPP Group Profile to a Tunnel

On Mi7 and M10i routers, you can optionally apply a configured *PPP* group profile to a tunnel. For any tunnel client, you can use the `user-group-profile` statement to define default PPP attributes for all users coming in through a tunnel. The user group profile must define PPP attributes. If the user group profile is specified, all users (PPP sessions) use the PPP attributes specified in the user group profile.

When a PPP client enters a tunnel, the Junos OS first applies the PPP user group profile attributes and then any PPP attributes from the local or *RADIUS* server. The PPP attributes defined in the *RADIUS* or local server take precedence over the attributes defined in the user group profile.

To apply configured PPP attributes to a PPP client, include the `user-group-profile` statement at the `[edit access profile profile-name client client-name]` hierarchy level:

```
[edit access profile profile-name client client-name]
user-group-profile profile-name;
```

*profile-name* is a PPP group profile configured at the [edit access group-profile *profile-name*] hierarchy level. When a client enters this tunnel, it uses the user-group-profile attributes as the default attributes.

Use a wildcard client to define a user group profile:

```
[edit access profile profile-name]
client * {
    user-group-profile profile-name;
}
```

## Compress PPP Fields

### IN THIS SECTION

- [Address and Control Field Compression | 15](#)
- [Protocol Field Compression | 16](#)
- [Monitor PPP Field Compression | 17](#)

By default, PPP does not compress the Layer 2 address, control, and protocol fields. Compressing these fields conserves bandwidth by transmitting less data. For interfaces with PPP, PPP CCC, or PPP TCC encapsulation, the device can compress the Layer 2 address, control, and protocol fields, as defined in RFC 1661, *The Point-to-Point Protocol (PPP)*.

Keep the following in mind when you configure PPP field compression:

- The PPP session restarts when you configure or modify compression options.
- The address, control, and protocol fields cannot be compressed in Link Control Protocol (LCP) packets.

### Address and Control Field Compression

Use address and control field compression (ACFC) to conserve bandwidth by transmitting less data. By default, the address and control fields of PPP-encapsulated packets are not compressed. This means PPP-encapsulated packets are transmitted with two one-byte fields (0xff and 0x03). If you configure ACFC and ACFC is successfully negotiated with the device's peer, the device transmits packets without these two bytes.

**NOTE:** On M320, M120, and T Series routers, ACFC is not supported for any ISO family protocols. Do not include the `acfc` statement at the `[edit interfaces interface-name ppp-options compression]` hierarchy level when you include the `family iso` statement at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level.

To configure ACFC:

1. In configuration mode, go to the `[edit interfaces interface-name ppp-options]` hierarchy level.

```
[edit]
user@host# edit interfaces interface-name ppp-options
```

2. Include the compression statement at the `[edit interfaces interface-name ppp-options]` hierarchy level and specify `acfc`.

```
[edit interfaces interface-name ppp-options]
user@device# set compression acfc;
```

## Protocol Field Compression

Use Protocol Field Compression (PFC) to compresses the protocol field of PPP-encapsulated packets. PFC conserves bandwidth by transmitting less data. By default, PPP-encapsulated packets are transmitted with a two-byte uncompressed protocol field. For example, IPv4 packets are transmitted with the protocol field set to 0x0021, and MPLS packets are transmitted with the protocol field set to 0x0281. For all protocols with identifiers in the range 0x0000 through 0x00ff, you can configure the router to compress the protocol field to one byte.

To configure PFC:

1. In configuration mode, go to the `[edit interfaces interface-name ppp-options]` hierarchy level.

```
[edit]
user@host# edit interfaces interface-name ppp-options
```

2. Include the `compression` statement at the `[edit interfaces interface-name ppp-options]` hierarchy level and specify `pfc`.

```
[edit interfaces interface-name ppp-options]
user@device# set compression pfc;
```

## Monitor PPP Field Compression

When ACFC and PFC are configured, the local device tries to negotiate ACFC and PFC with its peer. When you include the `compression` statement in the configuration, the PPP session restarts, and the local router sends the ACFC and PFC options in the LCP Configure-Request packet. The ACFC and PFC options inform the local router's peer that the local router can receive packets with compression.

If the peer indicates that it, too, can receive packets with compression, then ACFC and PFC are negotiated. If ACFC is successfully negotiated, the local router can receive packets with or without the address and control bytes included. If PFC is successfully negotiated, the local device can receive packets with either 2-byte (uncompressed) or 1-byte (compressed) protocol fields.

To monitor whether negotiation was successful, issue the `show interfaces interface-name` command. Configured options are displayed in the `Link flags` field for the physical interface. Successfully negotiated options are displayed in the `flags` field for the logical interface.

In this example, both ACFC and PFC are configured, but neither compression feature has been successfully negotiated:

```
user@device# run show interfaces so-0/1/1

Physical interface: so-0/1/1, Enabled, Physical link is Up
  Interface index: 133, SNMP ifIndex: 27
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC3, Loopback: None,
  FCS: 16,
  Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps 16384
  Link flags     : No-Keepalives ACFC PFC
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls: Not-configured
  CHAP state: Not-configured
  CoS queues    : 4 supported
  Last flapped  : 2004-12-29 10:49:32 PST (00:18:35 ago)
  Input rate    : 0 bps (0 pps)
  Output rate   : 0 bps (0 pps)
```

```

SONET alarms    : None
SONET defects   : None
Logical interface so-0/1/1.0 (Index 68) (SNMP ifIndex 169)
  Flags: Point-To-Point SNMP-Traps ACFC Encapsulation: PPP
  Protocol inet, MTU: 4470
    Flags: None
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 198.51.100/24, Local: 198.51.100.2, Broadcast: 198.51.100.255

```

## RELATED DOCUMENTATION

*ppp-options*

*compression*

*acfc*

*pfc*

## Configure PPP on ACX Series Routers

### IN THIS SECTION

- [Prepare Interfaces | 19](#)
- [Configure PPP Encapsulation | 19](#)

You can configure PPP encapsulation on physical interfaces on ACX Series routers. PPP is supported on the following MICs on ACX Series routers:

- On ACX1000 routers with 8-port built-in T1/E1 TDM MICs.
- On ACX2000, ACX2100, ACX2200, and ACX4000 routers with 16-port built-in T1/E1 TDM MICs.
- On ACX4000 routers with 16-Port Channelized E1/T1 Circuit Emulation MICs.
- Starting with Release 12.3X54, you can configure Point-to-Point Protocol (PPP) encapsulation on physical interfaces on Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP on ACX4000 Series routers.



## Prepare Interfaces

On ACX Series routers, E1, T1, and NxDS0 interfaces support PPP encapsulation. You might need to configure the interface before you can enable PPP encapsulation for that interface.

1. For full T1/E1 interfaces on which PPP encapsulation needs to be enabled, create the T1/E1 interfaces out of channelized T1/E1 interfaces (CT1/CE1) by including the framing statement at the [edit chassis fpc *fpc-slot* pic *pic-slot*] hierarchy level:

```
[edit chassis fpc fpc-slot pic pic-slot]
user@device# set framing (t1 | e1);
```

2. Configure a CT1 port down to a T1 channel. On the CT1 interface, set the no-partition option and then set the interface type as T1.

```
[edit interfaces ct1-mpc-slot/mic-slot/port-number]
user@device# set no-partition interface-type t1
```

3. Configure a CE1 port down to an E1 channel. On the CE1 interface, set the no-partition option and then set the interface type as E1.

```
[edit interfaces ce1-mpc-slot/mic-slot/port-number]
user@device# set no-partition interface-type e1
```

4. For NxDS0 interfaces on which PPP encapsulation needs to be enabled, partition the CE1 and CT1 interfaces:

```
[edit interfaces interface-name]
user@device# set ce1-x/y/z partition partition-number timeslots timeslots interface-type ds
set ct1-x/y/z partition partition-number timeslots timeslots interface-type ds
```

## Configure PPP Encapsulation

1. To configure the encapsulation on a physical interface, include the encapsulation ppp statement at the [edit interfaces *interface-name*] hierarchy level.
2. (Optional) On interfaces with PPP encapsulation, configure PPP-specific interface properties by including the ppp-options statement at the [edit interfaces *interface-name*] hierarchy level.
3. (Optional) PPP is supported only for IPv4 networks. You can configure the INET family by including the family inet statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.
4. (Optional) You can configure interfaces with PPP encapsulation to support the PPP Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP).

## Monitor a PPP Session

When PPP session monitoring is enabled, the operating system logs packets that are exchanged during a PPP session. It logs these packets to `/var/log/pppd` by default, or to the file specified in the `traceoptions` statement. When monitoring is configured, the operational mode commands `show ppp summary` and `show ppp interface` display a `Monitored` flag in the `Session flags` column or line.

To configure PPP session monitoring:

1. In configuration mode, go to the `[edit protocols ppp]` hierarchy level.

```
[edit]
user@host# edit protocols ppp
```

2. Include the `monitor-session` statement. You can monitor PPP packet exchanges on all PPP sessions or on a single logical interface.

```
[edit protocols ppp]
user@host# set monitor-session (interface-name | all)
```

## Monitor the PPP Process

You can monitor the operations of the device's PPP process (`pppd`) with the `traceoptions` statement. To monitor the device's `pppd`:

1. In configuration mode, go to the `[edit protocols ppp traceoptions]` hierarchy level.

```
[edit]
user@host# edit protocols ppp traceoptions
```

2. Configure the name of the file to receive the output of the tracing operation.

```
[edit protocols ppp traceoptions]
user@device# set file filename
```

3. Configure the tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.

```
[edit protocols ppp traceoptions]
user@device# set flag flag
```

## RELATED DOCUMENTATION

[Layer 2 Tunneling Protocol \(L2TP\) | 21](#)

*Logical Interface Properties*

*PPP Subscriber Access Networks Overview*

[Configuring MLPPP](#)

[Accessing Standards Documents on the Internet](#)

*ppp-options*

# Layer 2 Tunneling Protocol (L2TP)

## IN THIS SECTION

- [Minimum L2TP Configuration | 22](#)
- [L2TP Profiles | 23](#)
- [Example: L2TP Multilink PPP Support on Shared Interfaces | 30](#)
- [Example: PPP MP for L2TP | 31](#)
- [Configure L2TP Authentication | 32](#)
- [Example: Configure L2TP | 35](#)
- [Configure L2TP for M7i and M10i Routers | 37](#)

Layer 2 Tunneling Protocol (*L2TP*) is a protocol for tunneling Layer 2 traffic over a Layer 3 network. You can use L2TP to enable Point-to-Point Protocol (*PPP*) tunneling within your network.

L2TP requires an L2TP access concentrator (*LAC*) and an L2TP network server (*LNS*). The LNS is one endpoint of an L2TP tunnel. The LAC, configured on an access device, receives packets from a remote client and forwards them to the LNS on a remote network. The LAC and LNS are peers.

For information about how to configure L2TP for subscriber access , see *L2TP for Subscriber Access Overview*.

## Minimum L2TP Configuration

To define the minimum configuration for L2TP, include at least the following statements at the [edit access] hierarchy level:

```
[edit access]
address-pool pool-name {
    address address-or-prefix;
    address-range low <lower-limit> high <upper-limit>;
}
profile profile-name {
    authentication-order [ authentication-methods ];
    client client-name {
        chap-secret chap-secret;
        l2tp {
            interface-id interface-id;
            maximum-sessions-per-tunnel number;
            ppp-authentication (chap | pap);
            shared-secret shared-secret;
        }
        pap-password pap-password;
        ppp {
            framed-ip-address ip-address;
            framed-pool framed-pool;
            interface-id interface-id;
            primary-dns primary-dns;
            primary-wins primary-wins;
            secondary-dns secondary-dns;
            secondary-wins secondary-wins;
        }
    }
}
radius-server server-address {
```

```

accounting-port port-number;
port port-number;
retry attempts;
secret password;
}

```

**NOTE:** When the LNS is configured with *RADIUS* authentication, the default behavior is to accept the preferred RADIUS-assigned *IP* address. Previously, the default behavior was to accept and install the nonzero peer IP address received in the Internet Protocol Control Protocol (*IPCP*) configuration request packet.

## L2TP Profiles

### IN THIS SECTION

- [Access Profiles | 24](#)
- [Group Profile | 29](#)
- [Reference the Group Profile from the L2TP Profile | 30](#)

Configure profiles for L2TP.

When multiple types of profiles are configured, the operating system only implements one configuration. The operating system prioritizes them as follows, where [edit access profile *profile-name*] overrides all other profile configurations:

1. [edit access profile *profile-name*]
2. [edit access group-profile *profile-name*]
3. [edit access profile *profile-name* user-group-profile *profile-name*]

## Access Profiles

### IN THIS SECTION

- [Configure the L2TP Client | 24](#)
- [Client-Specific Profile | 24](#)
- [Example: Define the Default Tunnel Client | 26](#)
- [Configure the Access Profile | 27](#)
- [Example: Access Profile Configuration | 28](#)

To validate L2TP connections and session requests, you set up access profiles by configuring the profile statement at the [edit access] hierarchy level. You can configure multiple profiles. You can also configure multiple clients for each profile.

### Configure the L2TP Client

To configure the client, include the client statement at the [edit access profile *profile-name*] hierarchy level:

```
[edit access profile profile-name]
client client-name;
```

*client-name* is the peer identity.

For L2TP, you can optionally use the wildcard (\*) to define a default tunnel client to authenticate multiple LACs with the same secret and L2TP attributes. If an LAC with a specific name is not defined in the configuration, the wildcard tunnel client authenticates it.

**NOTE:** The \* for the default client configuration applies only to M Series routers. On MX Series routers, use default instead. See *L2TP LNS Inline Service Interfaces* for more about MX Series routers.

### Client-Specific Profile

To define *L2TP* properties for a client-specific profile, include one or more of the following statements at the [edit access profile *profile-name* client *client-name* l2tp] hierarchy level:

**NOTE:** When you configure the profile, you can configure either L2TP or *PPP* parameters, but not both at the same time.

```
[edit access profile profile-name client client-name l2tp]
interface-id interface-id;
lcp-renegotiation;
local-chap;
maximum-sessions-per-tunnel number;
multilink {
    drop-timeout milliseconds;
    fragment-threshold bytes;
}
ppp-authentication (chap | pap);
shared-secret shared-secret;
```

*interface-id* (in the *interface-id* statement) is the identifier for the interface representing an L2TP session configured at the [edit interfaces *interface-name* unit *local-unit-number* dial-options] hierarchy level.

*number* (in the *maximum-sessions-per-tunnel* statement) is the maximum number of sessions for an L2TP tunnel.

*shared-secret* (in the *shared-secret* statement) is the shared secret for authenticating the peer.

You can specify PPP authentication (in the *ppp-authentication* statement). By default, the PPP authentication uses CHAP. You can configure this to use Password Authentication Protocol (PAP).

You can configure LNS so it renegotiates LCP with the PPP client (in the *lcp-negotiation* statement). By default, the PPP client negotiates the LCP with the LAC. When you do this, the LNS discards the last sent LCP configuration request and last received LCP configuration request *AVPs* from the LAC; for example, the LCP negotiated between the PPP client and LAC.

You can configure the Junos OS so that the LNS ignores proxy authentication *AVPs* from the LAC and reauthenticates the PPP client using a CHAP challenge (in the *local-chap* statement). By default, the PPP client is not reauthenticated by the LNS. When you do this, the LNS directly authenticates the PPP client.

You can configure the PPP MP for L2TP if the PPP sessions that are coming into the LNS from the LAC have multilink PPP negotiated. When you do this, you join multilink bundles based on the endpoint discriminator (in the *multilink* statement).

- *milliseconds* (in the *drop-timeout* statement) specifies the number of milliseconds for the timeout that associated with the first fragment on the reassembly queue. If the timeout expires before all the

fragments have been collected, the fragments at the beginning of the reassembly queue are dropped. If the drop timeout is not specified, the Junos OS holds on to the fragments (fragments may still be dropped if the multilink reassembly algorithm determines that another fragment belonging to the packet on a reassembly queue has been lost).

**NOTE:** The drop timeout and fragmentation threshold for a bundled multilink might belong to different tunnels. The different tunnels might have different drop timeout and fragmentation thresholds. We recommend configuring group profiles instead of profiles when you have L2TP tunnels.

- *bytes* specifies the maximum size of a packet, in bytes (in the *fragment-threshold* statement). If a packet exceeds the fragmentation threshold, the Junos OS fragments it into two or more multilink fragments.

### Example: Define the Default Tunnel Client

```
[edit access profile profile-name]
client * {
  l2tp {
    interface-id interface1;
    lcp-renegotiation;
    local-chap;
    maximum-sessions-per-tunnel 500;
    ppp-authentication chap;
    shared-secret "$ABC123";
  }
}
```

For any tunnel client, you can optionally use the user group profile to define default *PPP* attributes for all users coming in through a tunnel. The user group profile must define PPP attributes. If the user group profile is specified, all users (PPP sessions) use the PPP attributes specified in the user group profile. The PPP attributes specified in the local or *RADIUS* server take precedence over those specified in the user group profile.

Optionally, you can use a wildcard client to define a user group profile. When you do this, any client entering this tunnel uses the PPP attributes (defined user group profile attributes) as its default PPP attributes.



## Configure the Access Profile

**NOTE:** When you configure a profile, you can only configure either L2TP or PPP parameters. You cannot configure both at the same time.

1. To configure the access profile, include the profile statement at the [edit access] hierarchy level and assign a name to the profile:

```
[edit access]
user@device# set profile profile-name
```

2. To configure the L2TP properties for a profile, include the following statements at the [edit access profile *profile-name*] hierarchy level:

```
[edit access profile profile-name]
client client-name {
    group-profile profile-name;
    l2tp {
        interface-id interface-id;
        lcp-renegotiation;
        local-chap;
        maximum-sessions-per-tunnel number;
        ppp-authentication (chap | pap);
        shared-secret shared-secret;
    }
}
user-group-profile profile-name;
```

3. You can configure the order in which the operating system tries different methods to authenticate peers. For each access attempt, the software tries the first configured authentication method. Configure the authentication order using the `authentication-order` statement. If you do not include the `authentication-order` statement, the operating system verifies clients using password authentication.

```
[edit access profile profile-name]
user@device# set authentication-order [ authentication-methods ]
```

In *authentication-methods*, specify one or more of the following in the preferred order. When you configure the authentication methods for L2TP, only the first configured authentication method is used.

- radius—Verify the client using RADIUS authentication services.
- password—Verify the client using the information configured at the [edit access profile *profile-name* client *client-name*] hierarchy level.

For L2TP, RADIUS authentication servers are configured at the [edit access radius-server] hierarchy level. For more information about configuring RADIUS authentication servers, see ["Configuring RADIUS Authentication for L2TP" on page 57](#).

## SEE ALSO

| *profile*

## Example: Access Profile Configuration

The following example shows a configuration of an access profile:

```
[edit access]
profile westcoast_bldg_1 {
  client white {
    chap-secret "$ABC123";
    # SECRET-DATA
    ppp {
      idle-timeout 22;
      primary-dns 198.51.100.10;
      framed-ip-address 198.51.100.12/24;
    }
    group-profile westcoast_users;
  }
  client blue {
    chap-secret "$ABC123";
    # SECRET-DATA
    group-profile sunnyvale_users;
  }
  authentication-order password;
}
profile westcoast_bldg_1_tunnel {
  client test {
    l2tp {
      shared-secret "$ABC123";
      # SECRET-DATA
      maximum-sessions-per-tunnel 75;
```

```

        ppp-authentication chap;
    }
    group-profile westcoast_tunnel;
}
client production {
    l2tp {
        shared-secret "$ABC123";
        # SECRET-DATA
        ppp-authentication chap;
    }
    group-profile westcoast_tunnel;
}
}

```

## Group Profile

You can optionally configure a group profile. Any client referencing the configured group profile inherits all the group profile attributes. This makes it easier to apply L2TP on a larger scale.

To configure the L2TP for the group profile, include the following statements at the [edit access group-profile *profile-name* l2tp] hierarchy level:

```

[edit access group-profile profile-name l2tp]
interface-id interface-id;
lcp-renegotiation;
local-chap;
maximum-sessions-per-tunnel number;

```

*interface-id* is the identifier for the interface representing an L2TP session configured at the [edit interfaces *interface-name* unit *local-unit-number* dial-options] hierarchy level.

You can configure the LNS so that it renegotiates the link control protocol (LCP) with the PPP client (in the renegotiation statement). By default, the PPP client negotiates the LCP with the L2TP access concentrator (LAC). When you do this, the LNS discards the last sent and the last received LCP configuration request attribute value pairs (AVPs) from the LAC; for example, the LCP negotiated between the PPP client and the LAC.

You can configure Junos OS so that the LNS ignores proxy authentication AVPs from the LAC and reauthenticates the PPP client using a CHAP challenge (in the local-chap statement). When you do this, the LNS directly authenticates the PPP client. By default, the PPP client is not reauthenticated by the LNS.

*number* is the maximum number of sessions per L2TP tunnel.

## Reference the Group Profile from the L2TP Profile

You can reference a configured group profile from the *L2TP* tunnel profile.

To reference the group profile configured at the [edit access group-profile *profile-name*] hierarchy level, include the group-profile statement at the [edit access profile *profile-name* client *client-name*] hierarchy level:

```
[edit access profile profile-name client client-name]
group-profile profile-name;
```

*profile-name* references a configured group profile from a PPP user profile.

## Example: L2TP Multilink PPP Support on Shared Interfaces

```
[edit]
interfaces {
  sp-1/3/0 {
    traceoptions {
      flag all;
    }
    unit 0 {
      family inet;
    }
    unit 20 {
      dial-options {
        l2tp-interface-id test;
        shared;
      }
      family inet;
    }
  }
}
access {
  profile t {
    client cholera {
      l2tp {
        interface-id test;
        multilink;
        shared-secret "$ABC123"; # SECRET-DATA
```

```

    }
  }
}
profile u {
  authentication-order radius;
}
radius-server {
  192.168.65.63 {
    port 1812;
    secret "$ABC123"; # SECRET-DATA
  }
}
}
services {
  l2tp {
    tunnel-group 1 {
      tunnel-access-profile t;
      user-access-profile u;
      local-gateway {
        address 10.70.1.1;
      }
      service-interface sp-1/3/0;
    }
    traceoptions {
      flag all;
      debug-level packet-dump;
      filter {
        protocol l2tp;
        protocol ppp;
        protocol radius;
      }
    }
  }
}
}

```

## Example: PPP MP for L2TP

```

[edit access]
profile tunnel-profile {

```

```

client remote-host {
    l2tp {
        multilink {
            drop-timeout 600;
            fragmentation-threshold 100;
        }
    }
}

```

## Configure L2TP Authentication

### IN THIS SECTION

- [Configure the CHAP Secret for an L2TP Profile | 32](#)
- [Example: Configuring L2TP PPP CHAP | 33](#)
- [Configure the PAP Password for an L2TP Profile | 34](#)
- [Example: Configure PAP for an L2TP Profile | 34](#)

L2TP does not include any authentication methods, so it is flexible and can be used with your preferred security features. When you configure *PPP* properties for an *L2TP* profile, you typically configure the `chap-secret` statement or `pap-password` statement.

### Configure the CHAP Secret for an L2TP Profile

*CHAP* allows each end of a *PPP* link to authenticate its peer, as defined in RFC 1994. The authenticator sends its peer a randomly generated challenge that the peer must encrypt using a one-way hash; the peer must then respond with that encrypted result. The key to the hash is a secret known only to the authenticator and authenticated. When the response is received, the authenticator compares its calculated result with the peer's response. If they match, the peer is authenticated.

Each end of the link identifies itself to its peer by including its name in the CHAP challenge and response packets it sends to the peer. This name defaults to the local hostname, or you can explicitly set it using the `local-name` option. When a host receives a CHAP challenge or CHAP response packet on a particular interface, it uses the peer identity to look up the CHAP secret key to use.

**NOTE:** When you configure PPP properties for a Layer 2 Tunneling Protocol (*L2TP*) profile, you typically configure the `chap-secret` statement or `pap-password` statement.

To configure CHAP, include the profile statement and specify a profile name at the `[edit access]` hierarchy level:

```
[edit access]
profile profile-name {
  client client-name chap-secret data;
}
```

Then reference the CHAP profile name at the `[edit interfaces interface-name ppp-options chap]` hierarchy level.

You can configure multiple profiles. You can also configure multiple clients for each profile.

`profile` is the mapping between peer identifiers and CHAP secret keys. The identity of the peer contained in the CHAP challenge or response queries the profile for the secret key to use.

`client` is the peer identity.

`chap-secret secret` is the secret key associated with that peer.

### Example: Configuring L2TP PPP CHAP

```
[edit]
access {
  profile westcoast_bldg1 {
    client cpe-1 chap-secret "$ABC123";
    # SECRET-DATA
    client cpe-2 chap-secret "$ABC123";
    # SECRET-DATA
  }
}
```

## Configure the PAP Password for an L2TP Profile

To configure the Password Authentication Protocol (*PAP*) password, include the `pap-password` statement at the `[edit access profile profile-name client client-name]` hierarchy level:

```
[edit access profile profile-name client client-name]  
pap-password pap-password;
```

*pap-password* is the password for PAP.

## Example: Configure PAP for an L2TP Profile

```
[edit access]  
profile sunnyvale_bldg_2 {  
  client green {  
    pap-password "$ABC123";  
    ppp {  
      interface-id west;  
    }  
    group-profile sunnyvale_users;  
  }  
  client red {  
    chap-secret "$ABC123";  
    group-profile sunnyvale_users;  
  }  
  authentication-order radius;  
}  
profile Sunnyvale_bldg_1_tunnel {  
  client test {  
    l2tp {  
      shared-secret "$ABC123";  
      ppp-authentication pap;  
    }  
  }  
}
```



## Example: Configure L2TP

```
[edit]
access {
    address-pool customer_a {
        address 10.1.1.1/32;
    }
    address-pool customer_b {
        address-range low 10.2.2.2 high 10.2.3.2;
    }
    group-profile westcoast_users {
        ppp {
            framed-pool customer_a;
            idle-timeout 15;
            primary-dns 10.192.65.1;
            secondary-dns 10.192.65.2;
            primary-wins 10.192.65.3;
            secondary-wins 10.192.65.4;
            interface-id west;
        }
    }
    group-profile eastcoast_users {
        ppp {
            framed-pool customer_b;
            idle-timeout 20;
            primary-dns 10.192.65.5;
            secondary-dns 10.192.65.6;
            primary-wins 10.192.65.7;
            secondary-wins 10.192.65.8;
            interface-id east;
        }
    }
    group-profile westcoast_tunnel {
        l2tp {
            maximum-sessions-per-tunnel 100;
        }
    }
    group-profile east_tunnel {
        l2tp {
            maximum-sessions-per-tunnel 125;
        }
    }
}
```

```

}
profile westcoast_bldg_1 {
    client white {
        chap-secret "$ABC123";
        # SECRET-DATA
        ppp {
            idle-timeout 22;
            primary-dns 10.192.65.10;
            framed-ip-address 10.12.12.12/32;
        }
        group-profile westcoast_users;
    }
    client blue {
        chap-secret "$ABC123";
        # SECRET-DATA
        group-profile sunnyvale_users;
    }
    authentication-order password;
}
profile west-coast_bldg_2 {
    client red {
        pap-password "$ABC123";
        # SECRET-DATA
        ppp {
            idle-timeout 22;
            primary-dns 10.192.65.11;
            framed-ip-address 10.12.12.12/32;
        }
        group-profile westcoast_users;
    }
}
profile westcoast_bldg_1_tunnel {
    client test {
        l2tp {
            shared-secret "$ABC123";
            # SECRET-DATA
            maximum-sessions-per-tunnel 75;
            ppp-authentication chap;# The default for PPP authentication is CHAP.
        }
        group-profile westcoast_tunnel;
    }
    client production {
        l2tp {

```

```

        shared-secret "$ABC123
        ABC123"; # SECRET-DATA
        ppp-authentication chap;
    }
    group-profile westcoast_tunnel;
}
}
profile westcoast_bldg_2_tunnel {
    client black {
        l2tp {
            shared-secret "$ABC123
            ABC123";
            # SECRET-DATA
            ppp-authentication pap;
        }
        group-profile westcoast_tunnel;
    }
}
}
}

```

## Configure L2TP for M7i and M10i Routers

For M7i and M10i routers, you can configure Layer 2 Tunneling Protocol (*L2TP*) tunneling security services on an Adaptive Services Physical Interface Card (*PIC*) or a MultiServices PIC.

To configure L2TP, include the following statements at the [edit access] hierarchy level:

```

[edit access]
address-pool pool-name {
    address address-or-prefix;
    address-range low <lower-limit> high <upper-limit>;
}
group-profile profile-name {
    l2tp {
        interface-id interface-id;
        lcp-renegotiation;
        local-chap;
        maximum-sessions-per-tunnel number;
        ppp {
            cell-overhead;

```

```

        encapsulation-overhead bytes;
        framed-pool pool-id;
        idle-timeout seconds;
        interface-id interface-id;
        keepalive seconds;
        primary-dns primary-dns;
        primary-wins primary-wins;
        secondary-dns secondary-dns;
        secondary-wins secondary-wins;
    }
}

profile profile-name {
    authentication-order [ authentication-methods ];
    accounting-order radius;
    client client-name {
        chap-secret chap-secret;
        group-profile profile-name;
        l2tp {
            interface-id interface-id;
            lcp-renegotiation;
            local-chap;
            maximum-sessions-per-tunnel number;
            ppp-authentication (chap | pap);
            shared-secret shared-secret;
        }
        pap-password pap-password;
        ppp {
            cell-overhead;
            encapsulation-overhead bytes;
            framed-ip-address ip-address;
            framed-pool framed-pool;
            idle-timeout seconds;
            interface-id interface-id;
            keepalive seconds;
            primary-dns primary-dns;
            primary-wins primary-wins;
            secondary-dns secondary-dns;
            secondary-wins secondary-wins;
        }
        user-group-profile profile-name;
    }
}

radius-disconnect-port port-number {

```

```

radius-disconnect {
    client-address {
        secret password;
    }
}

radius-server server-address {
    accounting-port port-number;
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    secret password;
    source-address source-address;
    timeout seconds;
}
}

```

#### RELATED DOCUMENTATION

[Address Pool for L2TP Network Server IP Address Allocation | 39](#)

[RADIUS Authentication for L2TP | 57](#)

## Address Pool for L2TP Network Server IP Address Allocation

With an *address pool*, you configure an address or address range. When you define an address pool for a client, the *L2TP* network server (*LNS*) allocates *IP* addresses for clients from an address pool. If you do not want to use an address pool, you can specify an IP address by means of the `framed-ip-address` statement at the `[edit access profile profile-name client client-name ppp]` hierarchy level. For information about specifying an IP address, see "[Point-to-Point Protocol \(PPP\)](#)" on page 4.

**NOTE:** When an address pool is modified or deleted, all the sessions using that pool are deleted.

To define an address or a range of addresses, include the `address-pool` statement at the `[edit access]` hierarchy level:

```
[edit access]
address-pool pool-name;
```

*pool-name* is the name assigned to the address pool.

To configure an address, include the `address` statement at the `[edit access address-pool pool-name]` hierarchy level:

```
[edit access address-pool pool-name]
address address-or-prefix;
```

*address-or-prefix* is one address or a prefix value.

When you specify an address range, it cannot exceed 65,535 IP addresses.

To configure the address range, include the `address-range` statement at the `[edit access address-pool pool-name]` hierarchy level:

```
[edit access address-pool pool-name]
address-range <low lower-limit> <high upper-limit>;
```

- `low lower-limit`—The lower limit of an address range.
- `high upper-limit`—The upper limit of an address range.

**NOTE:** The address pools for user access and Network Address Translation (NAT) can overlap. When you configure an address pool at the `[edit access address-pool pool-name]` hierarchy level, you can also configure an address pool at the `[edit services nat pool pool-name]` hierarchy level.

## RELATED DOCUMENTATION

[Point-to-Point Protocol \(PPP\) | 4](#)

[Layer 2 Tunneling Protocol \(L2TP\) | 21](#)

*Address-Assignment Pools for Subscriber Management*

# IKE Access Profiles

An Internet Key Exchange (*IKE*) access profile is used to negotiate IKE and *IPsec* security associations with *dynamic peers*. You can configure only one tunnel profile per service set for all dynamic peers. The configured *preshared key* in the profile is used for IKE authentication of all dynamic peers terminating in that service set. You can also use the digital certificate method for IKE authentication with dynamic peers. Include the `ike-policy policy-name` statement at the `[edit access profile profile-name client * ike]` hierarchy level. *policy-name* is the name of the IKE policy you define at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level.

The IKE tunnel profile specifies all the information you need to complete the IKE negotiation. Each protocol has its own statement hierarchy within the client statement to configure protocol-specific attribute value pairs, but only one client configuration is allowed for each profile. The following is the configuration hierarchy.

```
[edit access]
profile profile-name {
  client * {
    ike {
      allowed-proxy-pair {
        remote remote-proxy-address local local-proxy-address;
      }
      dead-peer-detection {
        interval seconds
        threshold number
      }
      ike-policy policy-name;
      initiate-dead-peer-detection;
      interface-id string-value;
      ipsec-policy ipsec-policy;
      pre-shared-key (ascii-text character-string | hexadecimal hexadecimal-digits);
      reverse-route
    }
  }
}
```

For dynamic peers, the Junos OS supports only IKE main mode with both the preshared key and digital certificate methods. In this mode, an IPv6 or IPv4 address is used to identify a tunnel peer to obtain the preshared key or digital certificate information. The client value `*` (wildcard) means that configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.

The following statement makes up the IKE profile:

- **allowed-proxy-pair**—During phase 2 IKE negotiation, the remote peer supplies its network address (*remote*) and its peer's network address (*local*). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

By default, *remote 0.0.0.0/0 local 0.0.0.0/0* is used if no values are configured.

- **dead-peer-detection**—Enable the device to use dead peer detection (DPD). DPD is a method used by devices to verify the current existence and availability of IPsec peer devices. A device performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE) to peers and waiting for DPD acknowledgements (R-U-THERE-ACK). Use the option *interval* to specify the seconds between which messages should be sent. Use the *threshold* option to specify the maximum number of messages (1-10) to be sent.
- **ike-policy**—Name of the IKE policy that defines either the local digital certificate or the preshared key used to authenticate the dynamic peer during IKE negotiation. You must include this statement to use the digital certificate method for IKE authentication with a dynamic peer. You define the IKE policy at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level.
- **initiate-dead-peer-detection**—Detects dead peers on dynamic IPsec tunnels.
- **interface-id**—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.
- **ipsec-policy**—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the `[edit services ipsec-vpn ipsec policy policy-name]` hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.
- **pre-shared-key**—Key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key is known to both ends through an out-of-band secure mechanism. You can configure the value either in hexadecimal or *ascii-text* format. It is a mandatory value.
- **reverse-route** —(M Series and MX Series routers with an AS or MultiServices PIC only) Configure a reverse route for dynamic endpoint IPsec tunnels.

## RELATED DOCUMENTATION



# 3

CHAPTER

## Configuring Authentication for PPP and L2TP

---

PPP Challenge Handshake Authentication Protocol | 44

Example: Configure CHAP Authentication with RADIUS | 49

PPP Password Authentication Protocol | 53

RADIUS Authentication for L2TP | 57

Subscriber Session Timeout Options | 74

---

# PPP Challenge Handshake Authentication Protocol

## IN THIS SECTION

- [PPP Challenge Handshake Authentication Protocol | 44](#)
- [Configuring the PPP Challenge Handshake Authentication Protocol | 44](#)
- [Displaying the Configured PPP Challenge Handshake Authentication Protocol | 47](#)
- [Example: Configuring PPP CHAP | 48](#)

## PPP Challenge Handshake Authentication Protocol

For interfaces with PPP encapsulation, you can configure interfaces to support the PPP Challenge Handshake Authentication Protocol (*CHAP*), as defined in RFC 1994, PPP Challenge Handshake Authentication Protocol (CHAP). When you enable CHAP on an interface, the interface can authenticate its peer and can be authenticated by its peer. By default, PPP CHAP is disabled. If CHAP is not explicitly enabled, the interface makes no CHAP challenges and denies all incoming CHAP challenges. To enable CHAP, you must create an access profile, and you must configure the interfaces to use CHAP.

CHAP allows each end of a PPP link to authenticate its peer, as defined in RFC 1994. The authenticator sends its peer a randomly-generated challenge that the peer must encrypt using a one-way *hash*; the peer must then respond with that encrypted result. The key to the hash is a secret known only to the authenticator and authenticated. When the response is received, the authenticator compares its calculated result with the peer's response. If they match, the peer is authenticated.

Each end of the link identifies itself to its peer by including its name in the CHAP challenge and response packets it sends to the peer. This name defaults to the local hostname, or you can explicitly set it using the `local-name` option. When a host receives a CHAP challenge or CHAP response packet on a particular interface, it uses the peer identity to look up the CHAP secret key to use.

## Configuring the PPP Challenge Handshake Authentication Protocol

To enable CHAP, you must create an access profile, and you must configure the interfaces to use PAP.

Definitions:

- `profile` is the mapping between peer identifiers and CHAP secret keys. The identity of the peer contained in the CHAP challenge or response queries the profile for the secret key to use.
- `client` is the peer identity.
- `chap-secret` is the secret key associated with that peer.

1. To create an access profile, include the `profile` statement at the `[edit access]` hierarchy level:

```
[edit access]
user@host# set profile profile-name {
```

2. To identify the peer and the secret key associated with that peer, include the `client` statement at the `[edit access profile profile-name]` hierarchy level:

```
[edit access profile profile-name]
user@host# set client client-name chap-secret chap-secret
```

You can configure multiple CHAP profiles, and configure multiple clients for each profile. For more information on how to configure access profile, see ["Point-to-Point Protocol \(PPP\)" on page 4](#) and ["Layer 2 Tunneling Protocol \(L2TP\)" on page 21](#).

When you configure an interface to use CHAP, you must assign an access profile to the interface. When an interface receives CHAP challenges and responses, the access profile in the packet is used to look up the shared secret, as defined in RFC 1994. If no matching access profile is found for the CHAP challenge that was received by the interface, the optionally configured default CHAP secret is used. The default CHAP secret is useful if the CHAP name of the peer is unknown, or if the CHAP name changes during PPP link negotiation.

To configure the PPP CHAP, on each physical interface with PPP encapsulation, perform the following steps.

1. To assign an access profile to an interface, include the `access-profile` statement at the `[edit interfaces interface-name ppp-options chap]` hierarchy level.

```
[edit interfaces interface-name ppp-options chap]
user@host# set access-profile name
```

**NOTE:** You must include the `access-profile` statement when you configure the CHAP authentication method. If an interface receives a CHAP challenge or response from a peer that is not in the applied access profile, the link is immediately dropped unless a default CHAP secret has been configured.

2. The default CHAP secret is used when no matching CHAP access profile exists, or if the CHAP name changes during PPP link negotiation. To configure a default CHAP secret for an interface, include the `default-chap-secret` statement at the `[edit interfaces interface-name ppp-options chap]` hierarchy level.

```
[edit interfaces interface-name ppp-options chap]
user@host# set default-chap-secret name
```

3. To configure the name the interface uses in CHAP challenge and response packets, include the `local-name` statement at the `[edit interfaces interface-name ppp-options chap]` hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
user@host# set local-name name
```

**NOTE:**

- The local name is any string from 1 through 32 characters in length, starting with an alphanumeric or underscore character, and including only the following characters:

a-z A-Z 0-9 % @ # / \ . \_ -

- By default, when CHAP is enabled on an interface, the interface uses the router's system hostname as the name sent in CHAP challenge and response packets.

4. You can configure the interface not to challenge its peer, and only respond when challenged. To configure the interface not to challenge its peer, include the `passive` statement at the `[edit interfaces interface-name ppp-options chap]` hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
user@host# set passive;
```

**NOTE:** By default, when CHAP is enabled on an interface, the interface always challenges its peer and responds to challenges from its peer.

## Displaying the Configured PPP Challenge Handshake Authentication Protocol

### IN THIS SECTION

- Purpose | 47
- Action | 47
- Meaning | 48

### Purpose

To display the configured PPP CHAP at the [edit access] and [edit interfaces] hierarchy levels.

- Access profile—pe-A-ppp-clients
- default CHAP secret data—"\$ABC123"
- hostname for the CHAP challenge and response packets—"pe-A-so-1/1/1"
- Interface—so-1/1/2

### Action

- Run the show command at the [edit access] hierarchy level.

```
profile pe-A-ppp-clients;  
client cpe-1 chap-secret "$ABC123";  
                # SECRET-DATA  
[edit interfaces so-1/2/0]  
encapsulation ppp;  
ppp-options {
```

```

    chap {
        access-profile pe-A-ppp-clients;
        default-chap-secret "$ABC123";
        local-name "pe-A-so-1/1/1";
    }
}

```

- Run the show command at the [edit interfaces so-1/1/2] hierarchy level.

```

ppp-options {
    chap {
        access-profile pe-A-ppp-clients;
        default-chap-secret "$ABC123";
        local-name "pe-A-so-1/1/2";
    }
}

```

## Meaning

The configured CHAP and its associated set options are displayed as expected.

## Example: Configuring PPP CHAP

```

[edit]
access {
    profile pe-A-ppp-clients {
        client cpe-1 chap-secret "$ABC123";
        # SECRET-DATA
        client cpe-2 chap-secret "$ABC123";
        # SECRET-DATA
    }
}
interfaces {
    so-1/1/1 {
        encapsulation ppp;
        ppp-options {
            chap {
                access-profile pe-A-ppp-clients;
            }
        }
    }
}

```

```
        local-name "pe-A-so-1/1/1";
    }
}
so-1/1/2 {
    encapsulation ppp;
    ppp-options {
        chap {
            passive;
            access-profile pe-A-ppp-clients;
            local-name "pe-A-so-1/1/2";
        }
    }
}
```

#### RELATED DOCUMENTATION

[Example: Configure CHAP Authentication with RADIUS | 49](#)

[PPP Password Authentication Protocol | 53](#)

## Example: Configure CHAP Authentication with RADIUS

#### IN THIS SECTION

- [Configuration | 50](#)

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 50](#)

You can send *RADIUS* messages through a routing instance to customer *RADIUS* servers in a private network. To configure the routing instance to send packets to a *RADIUS* server, include the `routing-instance statement` at the `[edit access profile profile-name radius-server]` hierarchy level and apply the profile to an interface with the `access-profile statement` at the `[edit interfaces interface-name unit logical-unit-number ppp-options chap]` hierarchy level.

In this example, *PPP* peers of interfaces `at-0/0/0.0` and `at-0/0/0.1` are authenticated by a *RADIUS* server reachable via routing instance A. *PPP* peers of interfaces `at-0/0/0.2` and `at-0/0/0.3` are authenticated by a *RADIUS* server reachable via routing instance B.

For more information about *RADIUS* authentication, see *RADIUS Authentication*.

### CLI Quick Configuration

```
system {
    radius-server {
        192.0.2.1 secret $ABC123;
        192.0.2.2 secret $ABC123;
    }
}
routing-instances {
    A {
        instance-type vrf;
        ...
    }
    B {
        instance-type vrf;
        ...
    }
}
access {
    profile A-PPP-clients {
```



```

authentication-order radius;
radius-server {
    192.0.2.3 {
        port 3333;
        secret "$ABC123"; # # SECRET-DATA
        timeout 3;
        retry 3;
        source-address 192.0.2.99;
        routing-instance A;
    }
    192.0.2.4 {
        routing-instance A;
        secret $ABC123;
    }
}
}
profile B-PPP-clients {
    authentication-order radius;
    radius-server {
        192.0.2.5 {
            routing-instance B;
            secret $ABC123;
        }
        192.0.2.6 {
            routing-instance B;
            secret $ABC123;
        }
    }
}
}
interfaces {
    at-0/0/0 {
        atm-options {
            vpi 0;
        }
        unit 0 {
            encapsulation atm-ppp-llc;
            ppp-options {
                chap {
                    access-profile A-PPP-clients;
                }
            }
        }
        keepalives {

```

```

        interval 20;
        up-count 5;
        down-count 5;
    }
    vci 0.128;
    family inet {
        address 192.0.2.21/32 {
            destination 192.0.2.22;
        }
    }
}
unit 1 {
    encapsulation atm-ppp-llc;
    ...
    ppp-options {
        chap {
            access-profile A-PPP-clients;
        }
    }
    ...
}
unit 2 {
    encapsulation atm-ppp-llc;
    ...
    ppp-options {
        chap {
            access-profile B-PPP-clients;
        }
    }
    ...
}
unit 3 {
    encapsulation atm-ppp-llc;
    ...
    ppp-options {
        chap {
            access-profile B-PPP-clients;
        }
    }
    ...
}
...
}

```

```
...
}
```

Users who log in to the router with *telnet* or *SSH* connections are authenticated by the RADIUS server 192.0.2.1. The backup RADIUS server for these users is 192.0.2.2.

Each profile may contain one or more backup RADIUS servers. In this example, PPP peers are *CHAP* authenticated by the RADIUS server 192.0.2.3 (with 192.0.2.4 as the backup server) or RADIUS server 192.0.2.5 (with 192.0.2.6 as the backup server).

## RELATED DOCUMENTATION

[Layer 2 Tunneling Protocol \(L2TP\) | 21](#)

[PPP Challenge Handshake Authentication Protocol | 44](#)

# PPP Password Authentication Protocol

## IN THIS SECTION

- [Understanding PAP | 53](#)
- [Configure PAP on a Physical Interface | 54](#)
- [Configure PAP on a Logical Interface | 55](#)

## Understanding PAP

The Password Authentication Protocol (PAP) provides a simple method for the peer to establish its identity using a two-way handshake. After the link is established, an ID and password pair is repeatedly sent by the peer to the authenticator until authentication is acknowledged or the connection is terminated. This is done only upon initial link establishment.

For interfaces with PPP encapsulation, you can configure interfaces to support the PAP, as defined in RFC 1334, *PAP Authentication Protocols*. If authentication is configured, the PPP link negotiates using CHAP or PAP protocol for authentication during the Link Control Protocol (LCP) negotiation phase. PAP is only performed after the link establishment phase (LCP up) portion of the authentication phase.

During authentication, the PPP link sends a PAP authentication-request packet to the peer with an ID and password. The authentication-request packet is sent every 2 seconds, similar to the CHAP challenge, until a response (acknowledgment packet or nonacknowledgment packet) is received. If an acknowledgment packet is received, the PPP link transitions to the next state, the network phase. If a nonacknowledgment packet is received, an LCP terminate request is sent, and the PPP link goes back to the link establishment phase.

If no response is received, and an optional retry counter is set to true, a new request acknowledgment packet is resent. If the retry counter expires, the PPP link transitions to the LCP negotiate phrase.

You can configure the PPP link with PAP in passive mode. By default, when PAP is enabled on an interface, the interface expects authenticate-request packets from the peer. However, the interface can be configured to send authentication request packets to the peer by configuring PAP to operate in passive mode. In PAP passive mode, the interface sends the authenticate-request packets to the peer only if the interface receives the PAP option from the peer during LCP negotiation. In passive mode, the interface does not authenticate the peer.

## Configure PAP on a Physical Interface

To enable PAP, you must create an access profile, and you must configure the interfaces to use PAP. For more information on how to configure access profile, see ["Point-to-Point Protocol \(PPP\)" on page 4](#).

When you configure an interface to use PAP, you must assign an access profile to the interface. When an interface receives PAP authentication requests, the access profile in the packet is used to look up the password.

To configure the PPP password authentication protocol, on each physical interface with PPP encapsulation, perform the following steps.

1. To assign an access profile to an interface, include the `access-profile` statement at the `[edit interfaces interface-name ppp-options pap]` hierarchy level.

```
[edit interfaces interface-name ppp-options pap]
user@host# set access-profile name
```

2. To configure the name the interface uses in PAP request and response packets, include the `local-name` statement at the `[edit interfaces interface-name ppp-options pap]` hierarchy level:

```
[edit interfaces interface-name ppp-options pap]
user@host# set local-name name
```

3. You need to configure the password to be used for authentication. To configure the host password for sending PAP requests, include the `local-password` statement at the `[edit interfaces interface-name ppp-options pap]` hierarchy level:

```
[edit interfaces interface-name ppp-options pap]
user@host# set local-password password
```

**NOTE:** By default, when PAP is enabled on an interface, the interface uses the router's system hostname as the name sent in PAP request and response packets.

4. To configure the interface to authenticate with PAP in passive mode, include the `passive` statement at the `[edit interfaces interface-name ppp-options pap]` hierarchy level:

```
[edit interfaces interface-name ppp-options pap]
user@host# set passive
```

**NOTE:** By default, when PAP is enabled on an interface, the interface expects authenticate-request packets from the peer. However, the interface can be configured to send authentication request packets to the peer by configuring PAP to operate in passive mode. In PAP passive mode, the interface sends the authenticate-request packets to the peer only if the interface receives the PAP option from the peer during LCP negotiation. In passive mode, the interface does not authenticate the peer.

## Configure PAP on a Logical Interface

When you configure an interface to use PAP, you must assign an access profile to the interface. When an interface receives PAP authentication requests, the access profile in the packet is used to look up the password. If no matching access profile is found for the PAP authentication request that was received by the interface, the optionally configured default PAP password is used.

To configure PAP, perform the following steps on each logical interface with PPP encapsulation.

1. The default PAP password is used when no matching PAP access profile exists, or if the PAP access profile name changes during PPP link negotiation. To configure the default PAP password, include the

default-pap-password statement at the [edit interfaces *interface-name* unit *logical-unit-number* ppp-options pap] hierarchy level:

```
[edit interfaces interface-name unit logical-unt-number ppp-options pap]
user@host# set default-pap-password password
```

2. To configure the name the interface uses in PAP request and response packets, include the `local-name` statement at the [edit interfaces *interface-name* unit *logical-unt-number* ppp-options pap] hierarchy level:

```
[edit interfaces interface-name ppp-options pap]
user@host# set local-name name
```

**NOTE:** By default, when PAP is enabled on an interface, the interface uses the router's system hostname as the name sent in PAP request and response packets.

3. You need to configure the password to be used for authentication. To configure the host password for sending PAP requests, include the `local-password` statement at the [edit interfaces *interface-name* ppp-options pap] hierarchy level:

```
[edit interfaces interface-name unit logical-unt-number ppp-options pap]
user@host# set local-password password
```

4. To configure the interface to authenticate with PAP in passive mode, include the `passive` statement at the [edit interfaces *interface-name* unit *logical-unt-number* ppp-options pap] hierarchy level:

```
[edit interfaces interface-name unit logical-unt-number ppp-options pap]
user@host# set passive
```

**NOTE:** By default, when PAP is enabled on an interface, the interface expects authenticate-request packets from the peer. However, the interface can be configured to send authentication request packets to the peer by configuring PAP to operate in passive mode. In PAP passive mode, the interface sends the authenticate-request packets to the peer only if the interface receives the PAP option from the peer during LCP negotiation—in passive mode, the interface does not authenticate the peer.

**SEE ALSO**

[PPP Challenge Handshake Authentication Protocol | 44](#)

## RADIUS Authentication for L2TP

**IN THIS SECTION**

- [Configure RADIUS Authentication for L2TP | 57](#)
- [Configure RADIUS Authentication for an L2TP Client and Profile | 59](#)
- [RADIUS Local Loopback Interface Attribute for L2TP | 60](#)
- [Example: Configure RADIUS Authentication for L2TP | 61](#)
- [Example: Configure RADIUS Authentication for an L2TP Profile | 62](#)
- [Configure the RADIUS Disconnect Server for L2TP | 63](#)
- [Configure RADIUS Accounting Order for L2TP | 64](#)
- [Example: Configure RADIUS-Based Subscriber Authentication and Accounting | 65](#)
- [RADIUS Attributes for L2TP | 68](#)

### Configure RADIUS Authentication for L2TP

The *L2TP* network server (*LNS*) sends *RADIUS* authentication requests or accounting requests. Authentication requests are sent out to the authentication server port. Accounting requests are sent to the accounting port. To configure RADIUS authentication for L2TP on an M10i or M7i router, include the following statements at the [edit access] hierarchy level:

```
[edit access]
radius-server server-address {
    accounting-port port-number;
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    secret password;
    source-address source-address;
```

```

    timeout seconds;
}

```

**NOTE:** The RADIUS servers at the [edit access] hierarchy level are not used by the network access server process (NASD).

You can specify an accounting port number on which to contact the accounting server (in the accounting-port statement). Most RADIUS servers use port number 1813 (as specified in RFC 2866, *Radius Accounting*).

**NOTE:** If you enable RADIUS accounting at the [edit access profile *profile-name* accounting-order] hierarchy level, accounting is triggered on the default port of 1813 even if you do not specify a value for the accounting-port statement.

*server-address* specifies the address of the RADIUS authentication server (in the radius-server statement).

You can specify a port number on which to contact the RADIUS authentication server (in the port statement). Most RADIUS servers use port number 1812 (as specified in RFC 2865, *Remote Authentication Dial In User Service [RADIUS]*).

You must specify a password in the secret statement. If a password includes spaces, enclose the password in quotation marks. The secret used by the local router must match that used by the RADIUS authentication server.

Optionally, you can specify the amount of time that the local router waits to receive a response from a RADIUS server (in the timeout statement) and the number of times that the router attempts to contact a RADIUS authentication server (in the retry statement). By default, the router waits 3 seconds. You can configure this to be a value in the range from 1 through 90 seconds. By default, the router retries connecting to the server three times. You can configure this to be a value in the range from 1 through 30 times. If the maximum number of retries is reached, the radius server is considered dead for 5 minutes (300 seconds).

In the source-address statement, specify a source address for each configured RADIUS server. Each RADIUS request sent to a RADIUS server uses the specified source address. The source address is a valid IPv4 address configured on one of the router interfaces.

To configure multiple RADIUS servers, include multiple radius-server statements.



**NOTE:** When the L2TP network server (LNS) is configured with RADIUS authentication, the default behavior is to accept the preferred RADIUS-assigned IP address. Previously, the default behavior was to accept and install the nonzero peer IP address received by the Internet Protocol Control Protocol (IPCP) configuration request packet.

## Configure RADIUS Authentication for an L2TP Client and Profile

On an M10i or M7i router, L2TP supports RADIUS authentication and accounting for users with one set of RADIUS servers under the [edit access] hierarchy. You can also configure RADIUS authentication for each tunnel client or user profile.

To configure the RADIUS authentication for L2TP tunnel clients on an M10i or M7i router, include the `ppp-profile` statement with the `l2tp` attributes for tunnel clients:

```
[edit access profile profile-name client client-name l2tp]
ppp-profile profile-name;
```

`ppp-profile profile-name` specifies the profile used to validate PPP session requests through L2TP tunnels. Clients of the referenced profile must have only PPP attributes. The referenced group profile must be defined.

To configure the RADIUS authentication for a profile, include following statements at the [edit access profile *profile-name*] hierarchy level:

```
[edit access profile profile-name]
radius-server server-address {
    accounting-port port-number;
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    secret password;
    source-address source-address;
    timeout seconds;
}
```

When a PPP user initiates a session and RADIUS authentication is configured for the user profile on the tunnel group, the following priority sequence is used to determine which RADIUS server is used for authentication and accounting:

- If the `ppp-profile` statement is configured under the tunnel client (LAC), the RADIUS servers configured under the specified `ppp-profile` are used.
- If RADIUS servers are configured under the user profile for the tunnel group, those servers will be used.
- If no RADIUS server is configured for the tunnel client (LAC) or user profile, then the RADIUS servers configured at the `[edit access]` hierarchy level are used.

## RADIUS Local Loopback Interface Attribute for L2TP

You can configure the Local-Loopback-Interface attribute on a RADIUS server to manage multiple LAC devices. This attribute is used as the LAC source address on an LNS tunnel for PPPoE subscribers tunneled over L2TP.

When you use the Tunnel-Client-Endpoint attribute as the LAC source address, you must configure the Tunnel-Client-Endpoint attribute for each MX Series router that uses the same RADIUS server. Starting with this release you can use the Local-Loopback-Interface attribute, which needs to be configured only once. When the LAC initiates an Access-Request message to RADIUS for authentication, RADIUS returns the Local-Loopback-Interface attribute in the Access-Accept message. This attribute contains the name of the loopback interface, either as a generic interface name such as "lo0" or as a specific name like "lo0.0". The MX Series router then uses the configured loopback interface IP address as the source address during tunnel negotiation with the LNS.

**NOTE:** An MX Series router can act as the LAC and use any interface address on it as an L2TP tunnel source address. The source address can be dynamically assigned by RADIUS through the Tunnel-Client-Endpoint or Local-Loopback-Interface attribute. The tunnel source address can be statically configured on the MX Series router by using the L2TP tunnel profile. If RADIUS does not return the Tunnel-Client-Endpoint or Local-Loopback-Interface attribute, and if there is no corresponding L2TP tunnel profile configured on the MX Series router, then the L2TP tunnel fails to initiate because the router does not have a proper tunnel source address. In this case, the router can use the locally configured loopback address as the source address to successfully establish the L2TP tunnel.

## Example: Configure RADIUS Authentication for L2TP

### IN THIS SECTION

- [Configuration | 61](#)

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 61](#)

### CLI Quick Configuration

The following example shows how to configure RADIUS authentication for L2TP:

```
[edit access]
profile example_bldg {
  client client_1 {
    chap-secret "$ABC123";
    ppp {
      interface-id west;
    }
    group-profile example_users;
  }
  client client_2 {
    chap-secret "$ABC123";
    group-profile example_users;
  }
  authentication-order radius;
}
radius-server {
  198.51.100.213 {
    port 1812;
    accounting-port 1813;
```

```

        secret "$ABC123"; # SECRET-DATA
    }
    198.51.100.223 {
        port 1812;
        accounting-port 1813;
        secret "$ABC123"; # SECRET-DATA
    }
}
radius-disconnect-port 2500;
radius-disconnect {
    198.51.100.152 secret "$ABC123";
    # SECRET-DATA
    198.51.100.153 secret "$ABC123";
    # SECRET-DATA
    198.51.100.157 secret "$ABC123";
    # SECRET-DATA
    198.51.100.173 secret "$ABC123";
    # SECRET-DATA
}

```

## Example: Configure RADIUS Authentication for an L2TP Profile

### IN THIS SECTION

- [Configuration | 62](#)

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 63](#)

## CLI Quick Configuration

```
[edit access]
profile t {
  client LAC_A {
    l2tp {
      ppp-profile u;
    }
  }
}
profile u {
  client client_1 {
    ppp {
    }
  }
  198.51.100.5 {
    port 3333;
    secret $ABC123;
    source-address 198.51.100.1;
    retry 3;
    timeout 3;
  }
  198.51.100.6 secret $ABC123;
  198.51.100.7 secret $ABC123;
}
```

## Configure the RADIUS Disconnect Server for L2TP

To configure the RADIUS disconnect server to listen for disconnect requests from an administrator and process them, include the following statements at the [edit access] hierarchy level:

```
[edit access]
radius-disconnect-port port-number;
radius-disconnect {
  client-address {
    secret password;
  }
}
```

*port-number* is the server port to which the RADIUS client sends disconnect requests. The L2TP network server, which accepts these disconnect requests, is the server. You can specify a port number on which to contact the RADIUS disconnect server. Most RADIUS servers use port number 1700.

**NOTE:** The Junos OS accepts only disconnect requests from the client address configured at the [edit access radius-disconnect *client-address*] hierarchy level.

*client-address* is the host sending disconnect requests to the RADIUS server. The client address is a valid IP address configured on one of the router or switch interfaces.

*password* authenticates the RADIUS client. Passwords can contain spaces. The secret used by the local router must match that used by the server.

For information about how to configure RADIUS authentication for L2TP, see ["Configuring RADIUS Authentication for L2TP" on page 57](#).

The following example shows the statements to be included at the [edit access] hierarchy level to configure the RADIUS disconnect server:

```
[edit access]
radius-disconnect-port 1700;
radius-disconnect {
    198.51.100.153 secret "$ABC123";
    # SECRET-DATA
    198.51.100.162 secret "$ABC123";
    # SECRET-DATA
}
```

## Configure RADIUS Accounting Order for L2TP

You can configure *RADIUS* accounting for an L2TP profile. With RADIUS accounting enabled, Juniper devices can act as RADIUS clients. They can notify the RADIUS server about user activities such as software logins, configuration changes, and interactive commands. The framework for RADIUS accounting is described in RFC 2866.

To configure RADIUS accounting, include the accounting-order statement at the [edit access profile *profile-name*] hierarchy level:

```
[edit access profile profile-name]  
accounting-order radius;
```

When you enable RADIUS accounting for an L2TP profile, it applies to all the clients within that profile. You must enable RADIUS accounting on at least one L2TP profile for the RADIUS authentication server to send accounting stop and start messages.

**NOTE:** When you enable RADIUS accounting for an L2TP profile, you do not need to configure the accounting-port statement at the [edit access radius-server *server-address*] hierarchy level. When you enable RADIUS accounting for an L2TP profile, accounting is triggered on the default port of 1813.

For L2TP, RADIUS authentication servers are configured at the [edit access radius-server] hierarchy level.

## Example: Configure RADIUS-Based Subscriber Authentication and Accounting

### IN THIS SECTION

- [Configuration | 65](#)

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 66](#)

## CLI Quick Configuration

```
[edit access]
radius-server {
  198.51.100.250 {
    port 1812;
    accounting-port 1813;
    accounting-retry 6;
    accounting-timeout 20;
    retry 3;
    secret $ABC123$ABC123;
    source-address 198.51.100.100;
    timeout 45;
  }
  198.51.100.251 {
    port 1812;
    accounting-port 1813;
    accounting-retry 6;
    accounting-timeout 20;
    retry 3;
    secret $ABC123;
    source-address 198.51.100.100;
    timeout 30;
  }
  2001:DB8:0f101::2{
    port 1812;
    accounting-port 1813;
    accounting-retry 6;
    accounting-timeout 20;
    retry 4;
    secret $ABC123$ABC123$ABC123-;
    source-address 2001:DB8:0f101::1;
    timeout 20;
  }
}
profile isp-bos-metro-fiber-basic {
  authentication-order radius;
  accounting {
    order radius;
    accounting-stop-on-access-deny;
    accounting-stop-on-failure;
    immediate-update;
  }
}
```



```

statistics time;
update-interval 12;
wait-for-acct-on-ack;
send-acct-status-on-config-change;
}
radius {
authentication-server 198.51.100.251 198.51.100.252;
accounting-server 198.51.100.250 198.51.100.251;
options {
accounting-session-id-format decimal;
client-accounting-algorithm round-robin;
client-authentication-algorithm round-robin;
nas-identifier 56;
nas-port-id-delimiter %;
nas-port-id-format {
nas-identifier;
interface-description;
}
nas-port-type {
ethernet {
wireless-80211;
}
}
}
attributes {
ignore {
framed-ip-netmask;
}
exclude {
accounting-delay-time [accounting-start accounting-stop];
accounting-session-id [access-request accounting-on accounting-off
accounting-start accounting-stop];
dhcp-gi-address [access-request accounting-start accounting-stop];
dhcp-mac-address [access-request accounting-start accounting-stop];
nas-identifier [access-request accounting-start accounting-stop];
nas-port [accounting-start accounting-stop];
nas-port-id [accounting-start accounting-stop];
nas-port-type [access-request accounting-start accounting-stop];
}
}
}
}
}

```

```
[edit logical-systems isp-bos-metro-12 routing-instances isp-cmbrg-12-32]
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 198.51.100.100/24;
      }
    }
  }
  ge-0/0/0 {
    vlan-tagging;
    unit 0 {
      vlan-id 200;
      family inet {
        unnumbered-address lo0.0;
      }
    }
  }
}
}
```

## RADIUS Attributes for L2TP

Junos OS supports the following types of RADIUS attributes for L2TP:

- Juniper Networks vendor-specific attributes (VSAs)
- Attribute-value pairs (AVPs) defined by the Internet Engineering Task Force (IETF)
- RADIUS accounting stop and start AVPs

Juniper Networks vendor-specific RADIUS attributes are described in RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*. These attributes are encapsulated with the vendor ID set to the Juniper Networks ID number 2636. [Table 1 on page 68](#) lists the Juniper Networks VSAs you can configure for L2TP.

**Table 1: Juniper Networks Vendor-Specific RADIUS Attributes for L2TP**

Attribute Name	Standard Number	Value
Juniper-Primary-DNS	31	IP address

**Table 1: Juniper Networks Vendor-Specific RADIUS Attributes for L2TP (Continued)**

Attribute Name	Standard Number	Value
Juniper-Primary-WINS	32	IP address
Juniper-Secondary-DNS	33	IP address
Juniper-Secondary-WINS	34	IP address
Juniper-Interface-ID	35	String
Juniper-IP-Pool-Name	36	String
Juniper-Keep-Alive	37	Integer

[Table 2 on page 69](#) lists the IETF RADIUS AVPs supported for L2TP.

**Table 2: Supported IETF RADIUS Attributes for L2TP**

Attribute Name	Standard Number	Value
User-Name	1	String
User-Password	2	String
CHAP-Password	3	String
NAS-IP-Address	4	IP address
NAS-Port	5	Integer
Service-Type	6	Integer
Framed-Protocol	7	Integer

**Table 2: Supported IETF RADIUS Attributes for L2TP (Continued)**

Attribute Name	Standard Number	Value
Framed-IP-Address	8	IP address
Framed-IP-Netmask	9	IP address
Framed-MTU	12	Integer
Framed-Route	22	String
Session-Timeout	27	Integer
Idle-Timeout	28	Integer
Called-Station-ID	30	String
Calling-Station-ID	31	String
CHAP-Challenge	60	String
NAS-Port-Type	61	Integer
Framed-Pool	88	Integer

[Table 3 on page 70](#) lists the supported RADIUS accounting start AVPs for L2TP.

**Table 3: Supported RADIUS Accounting Start Attributes for L2TP**

Attribute Name	Standard Number	Value
User-Name	1	String
NAS-IP-Address	4	IP address

**Table 3: Supported RADIUS Accounting Start Attributes for L2TP (Continued)**

Attribute Name	Standard Number	Value
NAS-Port	5	Integer
Service-Type	6	Integer
Framed-Protocol	7	Integer
Framed-IP-Address	8	IP address
Called-Station-ID	30	String
Calling-Station-ID	31	String
Acct-Status-Type	40	Integer
Acct-Delay-Time	41	Integer
Acct-Session-ID	44	String
Acct-Authentic	45	Integer
NAS-Port-Type	61	Integer
Tunnel-Client-Endpoint	66	String
Tunnel-Server-Endpoint	67	String
Acct-Tunnel-Connection	68	String
Tunnel-Client-Auth-ID	90	String

**Table 3: Supported RADIUS Accounting Start Attributes for L2TP (Continued)**

Attribute Name	Standard Number	Value
Tunnel-Server-Auth-ID	91	String

[Table 4 on page 72](#) lists the supported RADIUS accounting stop AVPs for L2TP.

**Table 4: Supported RADIUS Accounting Stop Attributes for L2TP**

Attribute Name	Standard Number	Value
User-Name	1	String
Local-Loopback-Interface	3	String
NAS-IP-Address	4	IP address
NAS-Port	5	Integer
Service-Type	6	Integer
Framed-Protocol	7	Integer
Framed-IP-Address	8	IP address
Called-Station-ID	30	String
Calling-Station-ID	31	String
Acct-Status-Type	40	Integer
Acct-Delay-Time	41	Integer
Acct-Input-Octets	42	Integer

**Table 4: Supported RADIUS Accounting Stop Attributes for L2TP (Continued)**

Attribute Name	Standard Number	Value
Acct-Output-Octets	43	Integer
Acct-Session-ID	44	String
Acct-Authentic	45	Integer
Acct-Session-Time	46	Integer
Acct-Input-Packets	47	Integer
Acct-Output-Packets	48	Integer
Acct-Terminate-Cause	49	Integer
Acct-Multi-Session-ID	50	String
Acct-Link-Count	51	Integer
NAS-Port-Type	61	Integer
Tunnel-Client-Endpoint	66	String
Tunnel-Server-Endpoint	67	String
Acct-Tunnel-Connection	68	String
Tunnel-Client-Auth-ID	90	String
Tunnel-Server-Auth-ID	91	String

## RELATED DOCUMENTATION

[Layer 2 Tunneling Protocol \(L2TP\) | 21](#)

## Subscriber Session Timeout Options

Subscriber session timeout options enable you to place limits on subscriber access based on how long the session has been up, how long the user has been inactive, or both. The subscriber session options apply to both L2TP-tunneled and PPP-terminated subscriber sessions. For DHCP subscribers, the session timeout limits the DHCP lease time.

**NOTE:** To configure the timeout attributes in RADIUS, refer to the documentation for your RADIUS server.

To configure limitations on subscriber sessions, configure the session options in the client profile that applies to the subscriber:

- Terminate the subscriber when the configured session timeout expires, regardless of activity.

```
[edit access profile profile-name session-options]  
user@host# set client-session-timeout minutes
```

- Terminate the subscriber when there is no ingress or egress data traffic for the duration of the configured idle timeout.

```
[edit access profile profile-name session-options]  
user@host# set client-idle-timeout minutes
```

- Terminate the subscriber when there is no ingress data traffic for the duration of the configured idle timeout; ignore egress traffic.

```
[edit access profile profile-name session-options]  
user@host# set client-idle-timeout minutes  
user@host# set client-idle-timeout-ingress-only
```



For example, to configure session timeout options in the `acc-prof` client profile, specifying an idle timeout of 15 minutes, that only ingress traffic is monitored, and that the session times out after 120 minutes:

```
[edit]
access {
  profile {
    acc-prof {
      session-options {
        client-idle-timeout 15;
        client-idle-timeout-ingress-only;
        client-session-timeout 120;
      }
    }
  }
}
```

# 4

CHAPTER

## Configuration Statements and Operational Commands

---

[Junos CLI Reference Overview](#) | 77

---

# Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax, options, and release information that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- *Junos CLI Reference*

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- *Configuration Statements*
- *CLI Commands*