

Junos® OS

Broadband Subscriber Access Protocols User Guide

Published
2023-12-18

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
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 Broadband Subscriber 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 | xii

1

Broadband Subscriber Access Network Overview

Broadband Subscriber Access Network Overview | 2

Subscriber Access Network Overview | 2

Multiservice Access Node Overview | 3

Ethernet MSAN Aggregation Options | 5

LDP Pseudowire Autosensing Overview | 7

Layer 2 Services on Pseudowire Service Interface Overview | 10

Broadband Access Service Delivery Options | 20

Broadband Delivery and FTTx | 22

Understanding BNG Support for Cascading DSLAM Deployments Over Bonded DSL Channels | 23

Detection of Backhaul Line Identifiers and Autogeneration of Intermediate Node Interface Sets | 27

High Availability for Subscriber Access Networks | 31

Unified ISSU for High Availability in Subscriber Access Networks | 32

Verifying and Monitoring Subscriber Management Unified ISSU State | 33

Graceful Routing Engine Switchover for Subscriber Access Networks | 34

Minimize Traffic Loss Due to Stale Route Removal After a Graceful Routing Engine Switchover | 35

Routes for DHCP and PPP Subscriber Access Networks | 37

Access and Access-Internal Routes for Subscriber Management | 37

Configuring Dynamic Access Routes for Subscriber Management | 38

Configuring Dynamic Access-Internal Routes for DHCP and PPP Subscribers | 40

Suppressing DHCP Access, Access-Internal, and Destination Routes | 41

Preventing DHCP from Installing Access, Access-Internal, and Destination Routes by Default | 42

Verifying the Configuration of Access and Access-Internal Routes for DHCP and PPP Subscribers | 43

Subscribers with Identical Framed Routes | 45

Configuring PWHT on a Transport Logical Interface for BNG | 46

Overview | 46

Requirements | 47

Configuration | 48

2

DHCP Subscriber Access Networks

DHCP Subscriber Access Networks Overview | 57

DHCP and Subscriber Management Overview | 57

Subscriber Access Operation Flow Using DHCP Relay | 59

Defining Various Levels of Services for DHCP Subscribers | 60

Example: Configuring a Tiered Service Profile for Subscriber Access | 61

DHCP Snooping for Network Security | 65

DHCP Snooping Support | 66

Configuring DHCP Snooped Packets Forwarding Support for DHCP Local Server | 69

Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent | 70

Configuring DHCP Snooped Packets Forwarding Support for DHCP Relay Agent | 77

Disabling DHCP Snooping Filters | 80

Example: Configuring DHCP Snooping Support for DHCP Relay Agent | 82

Requirements | 82

Overview | 82

Configuration | 82

Example: Enabling DHCP Snooping Support for DHCPv6 Relay Agent | 85

Requirements | 85

Overview | 85

Configuration | 86

Verification | 89

Preventing DHCP Spoofing | 90

DHCPv4 Duplicate Client Management | 92

DHCPv4 Duplicate Client In Subnet Overview | 92

Guidelines for Configuring Support for DHCPv4 Duplicate Clients | 93

Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Option 82 Information | 94

Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Their Incoming Interfaces | 96

DHCPv6 Duplicate Client Management | 98

DHCPv6 Duplicate Client DUIDs | 98

Configuring the Router to Use Underlying Interfaces to Distinguish Between DHCPv6 Duplicate Client DUIDs | 99

3

PPP Subscriber Access Networks

PPP Subscriber Access Networks Overview | 102

Dynamic Profiles for PPP Subscriber Interfaces Overview | 102

Understanding How the Router Processes Subscriber-Initiated PPP Fast Keepalive Requests | 103

RADIUS-Sourced Connection Status Updates to CPE Devices | 106

Configure Dynamic Profiles for PPP | 111

Preventing the Validation of PPP Magic Numbers During PPP Keepalive Exchanges | 112

How to Configure RADIUS-Sourced Connection Status Updates to CPE Devices | 113

Attaching Dynamic Profiles to Static PPP Subscriber Interfaces | 114

Migrating Static PPP Subscriber Configurations to Dynamic Profiles Overview | 115

Configuring Local Authentication in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers | 117

Configuring Tag2 Attributes in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers | 119

Configuring Dynamic Authentication for PPP Subscribers | 120

Modifying the CHAP Challenge Length | 122

Example: Minimum PPPoE Dynamic Profile | 124

Verifying and Managing PPP Configuration for Subscriber Management | 124

PPP Network Control Protocol Negotiation | 126

PPP Network Control Protocol Negotiation Mode Overview | 126

Controlling the Negotiation Order of PPP Authentication Protocols | 129

Configuring the PPP Network Control Protocol Negotiation Mode | 132

Ensuring IPCP Negotiation for Primary and Secondary DNS Addresses | 134

Tracing PPP Service Events for Troubleshooting | 136

Configuring the PPP Service Trace Log Filename | 137

Configuring the Number and Size of PPP Service Log Files | 138

Configuring Access to the PPP Service Log File | 138

Configuring a Regular Expression for PPP Service Messages to Be Logged | 139

Configuring Subscriber Filtering for PPP Service Trace Operations | 139

Configuring the PPP Service Tracing Flags | 141

Configuring the Severity Level to Filter Which PPP Service Messages Are Logged | 141

4

L2TP Subscriber Access Networks

L2TP for Subscriber Access Overview | 144

L2TP for Subscriber Access Overview | 144

L2TP Terminology | 147

L2TP Implementation | 148

Retransmission of L2TP Control Messages | 151

Configuring Retransmission Attributes for L2TP Control Messages | 152

Enabling Tunnel and Global Counters for SNMP Statistics Collection | 154

Verifying and Managing L2TP for Subscriber Access | 155

L2TP Tunnel Switching For Multiple-Domain Networks | 158

L2TP Tunnel Switching Overview | 158

Tunnel Switching Actions for L2TP AVPs at the Switching Boundary | 163

- Configuring L2TP Tunnel Switching | 168
- Setting the L2TP Receive Window Size | 171
- Setting the L2TP Tunnel Idle Timeout | 171
- Setting the L2TP Destruct Timeout | 172
- Configuring the L2TP Destination Lockout Timeout | 172
- Removing an L2TP Destination from the Destination Lockout List | 173
- Configuring L2TP Drain | 174
- Using the Same L2TP Tunnel for Injection and Duplication of IP Packets | 175

L2TP LAC Subscriber Configuration | 176

- Configuring an L2TP LAC | 176
- Configuring How the LAC Responds to Address and Port Changes Requested by the LNS | 177
- LAC Interoperation with Third-Party LNS Devices | 180
- Globally Configuring the LAC to Interoperate with Cisco LNS Devices | 181

L2TP LAC Tunneling for Subscribers | 182

- LAC Tunnel Selection Overview | 183
- L2TP Session Limits Overview | 200
- Limiting the Number of L2TP Sessions Allowed by the LAC or LNS | 206
- Setting the Format for the Tunnel Name | 209
- Configuring a Tunnel Profile for Subscriber Access | 210
- Configuring the L2TP LAC Tunnel Selection Parameters | 213
- Configuring LAC Tunnel Selection Failover Within a Preference Level | 213
- Configuring Weighted Load Balancing for LAC Tunnel Sessions | 214
- Configuring Destination-Equal Load Balancing for LAC Tunnel Sessions | 215
- Enabling the LAC for IPv6 Services | 215
- Testing L2TP Tunnel Configurations from the LAC | 216

L2TP Subscriber Access Lines and Connection Speeds | 219

Subscriber Access Line Information Handling by the LAC and LNS Overview	219
Transmission of Tx and Rx Connection Speeds from LAC to LNS	232
Transmission of the Receive Connect Speed AVP When Transmit and Receive Connect Speeds are Equal	241
Configuring the Method to Derive the LAC Connection Speeds Sent to the LNS	242
Configuring the Reporting and Processing of Subscriber Access Line Information	245
Preventing the LAC from Sending Calling Number AVP 22 to the LNS	250
Override the Calling-Station-ID Format for the Calling Number AVP	251
Specifying a Rate-Limiting Service Profile for L2TP Connection Speeds	253

L2TP LNS Inline Service Interfaces | 259

Configuring an L2TP LNS with Inline Service Interfaces	259
Applying PPP Attributes to L2TP LNS Subscribers per Inline Service Interface	261
Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile	264
Configuring an L2TP Access Profile on the LNS	266
Configuring a AAA Local Access Profile on the LNS	268
Configuring an Address-Assignment Pool for L2TP LNS with Inline Services	269
Configuring the L2TP LNS Peer Interface	271
Enabling Inline Service Interfaces	272
Configuring an Inline Service Interface for L2TP LNS	274
Configuring Options for the LNS Inline Services Logical Interface	275
LNS 1:1 Stateful Redundancy Overview	276
Configuring 1:1 LNS Stateful Redundancy on Aggregated Inline Service Interfaces	276
Verifying LNS Aggregated Inline Service Interface 1:1 Redundancy	279
L2TP Session Limits and Load Balancing for Service Interfaces	282
Example: Configuring an L2TP LNS	286
Requirements	286
Overview	287

| Configuration | 289

Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces | 301

Applying Services to an L2TP Session Without Using RADIUS | 304

Configuring a Pool of Inline Services Interfaces for Dynamic LNS Sessions | 313

Configuring a Dynamic Profile for Dynamic LNS Sessions | 314

IP Packet Reassembly on Inline Service Interfaces | 317

| IP Packet Fragment Reassembly for L2TP Overview | 318

| Configuring IP Inline Reassembly for L2TP | 320

Peer Resynchronization After an L2TP Failover | 322

| L2TP Failover and Peer Resynchronization | 323

| Configuring the L2TP Peer Resynchronization Method | 324

Tracing L2TP Events for Troubleshooting | 326

| Configuring the L2TP Trace Log Filename | 327

| Configuring the Number and Size of L2TP Log Files | 328

| Configuring Access to the L2TP Log File | 328

| Configuring a Regular Expression for L2TP Messages to Be Logged | 329

| Configuring Subscriber Filtering for L2TP Trace Operations | 329

| Configuring the L2TP Tracing Flags | 331

| Configuring the Severity Level to Filter Which L2TP Messages Are Logged | 331

5

Configuring MPLS Pseudowire Subscriber Logical Interfaces

MPLS Pseudowire Subscriber Logical Interfaces | 334

| Pseudowire Subscriber Logical Interfaces Overview | 334

| Anchor Redundancy Pseudowire Subscriber Logical Interfaces Overview | 338

| Configuring a Pseudowire Subscriber Logical Interface | 345

| Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router | 347

| Configuring a Pseudowire Subscriber Logical Interface Device | 348

- Changing the Anchor Point for a Pseudowire Subscriber Logical Interface Device | 350
- Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface | 353
- Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces | 354
- Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces | 355
- Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface | 357
- Configuring a PWHT with VC 11 Type Support | 359
- Configuring Load Balancing Support for Subscriber Traffic | 362

6

Wi-Fi Access Gateways

Wi-Fi Access Gateways | 372

- Wi-Fi Access Gateway Overview | 372
- Wi-Fi Access Gateway Deployment Model Overview | 374
- Supported Access Models for Dynamic-Bridged GRE Tunnels on the Wi-Fi Access Gateway | 376
- Wi-Fi Access Gateway Configuration Overview | 377
- Configuring a Pseudowire Subscriber Logical Interface Device for the Wi-Fi Access Gateway | 377
- Configuring Conditions for Enabling Dynamic-Bridged GRE Tunnel Creation | 379
- Configuring VLAN Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways | 382
- Configuring Untagged Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways | 387

7

Fixed Wireless Access Networks

Fixed Wireless Access Networks | 390

- Fixed Wireless Access Network Overview | 390
- How to Configure Fixed Wireless Access | 401
- Verifying and Monitoring Fixed Wireless Access | 406

Tracing Fixed Wireless Access Events for Troubleshooting | 407

- Configuring the Fixed Wireless Access Trace Log Filename | 408
- Configuring the Number and Size of Fixed Wireless Access Log Files | 408

Configuring Access to the Fixed Wireless Access Log File | 409

Configuring a Regular Expression for Fixed Wireless Access Messages to Be Logged | 409

Configuring the Fixed Wireless Access Tracing Flags | 410

8

Configuration Statements and Operational Commands

Junos CLI Reference Overview | 412

About This Guide

Use this guide to understand how to configure the primary methods for accessing the subscriber network:

- DHCP provides IP address configuration and service provisioning.
- PPP enables a point-to-point direct connection to the network and service provider. Dynamic profiles apply configurations and services to authenticated subscribers.
- L2TP separates the termination of access technologies from the termination of PPP and subsequent access to a network. This separation enables service providers to outsource their access technologies. L2TP provides ISPs the capability to supply VPN service; private enterprises can reduce or avoid investment in access technologies for remote workers.
- MPLS pseudowire interfaces extend MPLS domains from the access-aggregation network to the service edge.
- Wi-Fi access gateways provide public Wi-Fi access from residential or business Wi-Fi networks so that mobile subscribers can be authenticated and connected regardless of their physical location.
- Fixed wireless access enables service providers to manage subscribers over a wireless network to the home instead of having to run fiber to the building. The wireless network reduces last-mile installation and maintenance costs and gives providers the ability to increase services to underserved end users.

RELATED DOCUMENTATION

[Configuring the Broadband Edge as a Service Node Within Seamless MPLS Network Designs](#)

[Configuring MX Series Universal Edge Routers for Service Convergence](#)

1

CHAPTER

Broadband Subscriber Access Network Overview

Broadband Subscriber Access Network Overview | 2

High Availability for Subscriber Access Networks | 31

Routes for DHCP and PPP Subscriber Access Networks | 37

Subscribers with Identical Framed Routes | 45

Configuring PWHT on a Transport Logical Interface for BNG | 46

Broadband Subscriber Access Network Overview

IN THIS SECTION

- [Subscriber Access Network Overview | 2](#)
- [Multiservice Access Node Overview | 3](#)
- [Ethernet MSAN Aggregation Options | 5](#)
- [LDP Pseudowire Autosensing Overview | 7](#)
- [Layer 2 Services on Pseudowire Service Interface Overview | 10](#)
- [Broadband Access Service Delivery Options | 20](#)
- [Broadband Delivery and FTTx | 22](#)
- [Understanding BNG Support for Cascading DSLAM Deployments Over Bonded DSL Channels | 23](#)
- [Detection of Backhaul Line Identifiers and Autogeneration of Intermediate Node Interface Sets | 27](#)

Subscriber Access Network Overview

A subscriber access environment can include various components, including subscriber access technologies and authentication protocols.

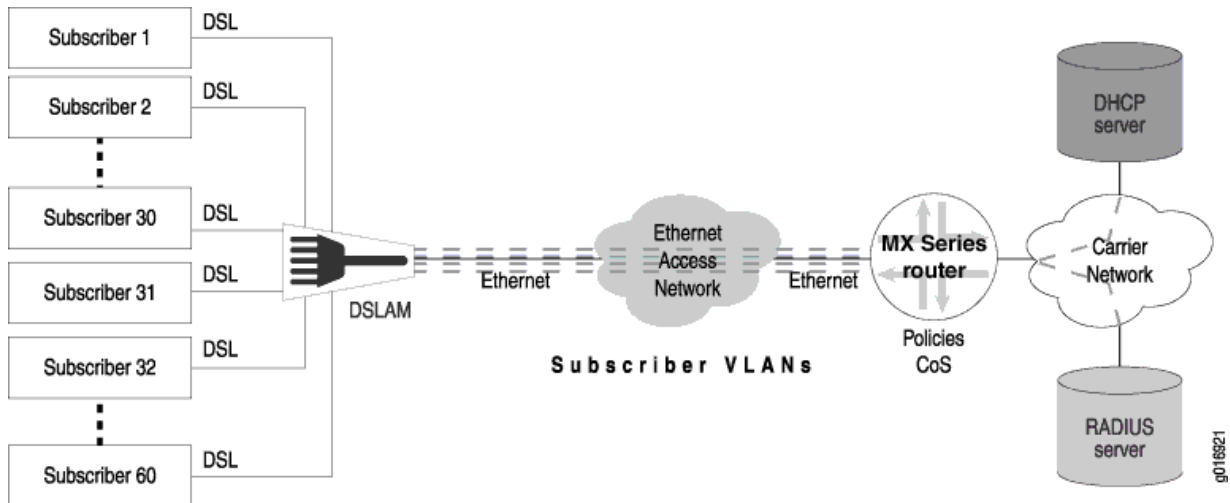
The subscriber access technologies include:

- Dynamic Host Configuration Protocol (DHCP) server
 - Local DHCP server
 - External DHCP server
- Point-to-Point Protocol (PPP)

The subscriber authentication protocols include the RADIUS server.

[Figure 1 on page 3](#) shows an example of a basic subscriber access network.

Figure 1: Subscriber Access Network Example



NOTE: This feature requires a license. To understand more about Subscriber Access Licensing, see, [Subscriber Access Licensing Overview](#). Please refer to the Juniper Licensing Guide for general information about License Management. Please refer to the product Data Sheets at [MX Series Routers](#) for details, or contact your Juniper Account Team or Juniper Partner.

Multiservice Access Node Overview

A *multiservice access node* is a broader term that refers to a group of commonly used aggregation devices. These devices include digital subscriber line access multiplexers (DSLAMs) used in xDSL networks, optical line termination (OLT) for PON/FTTx networks, and Ethernet switches for Active Ethernet connections. Modern MSANs often support all of these connections, as well as providing connections for additional circuits such as plain old telephone service (referred to as POTS) or Digital Signal 1 (DS1 or T1).

The defining function of a multiservice access node is to aggregate traffic from multiple subscribers. At the physical level, the MSAN also converts traffic from the *last mile technology* (for example, ADSL) to Ethernet for delivery to subscribers.

You can broadly categorize MSANs into three types based on how they forward traffic in the network:

- **Layer-2 MSAN**—This type of MSAN is essentially a Layer 2 switch (though typically not a fully functioning switch) with some relevant enhancements. These MSANs use Ethernet (or ATM) switching to forward traffic. The MSAN forwards all subscriber traffic upstream to an edge router

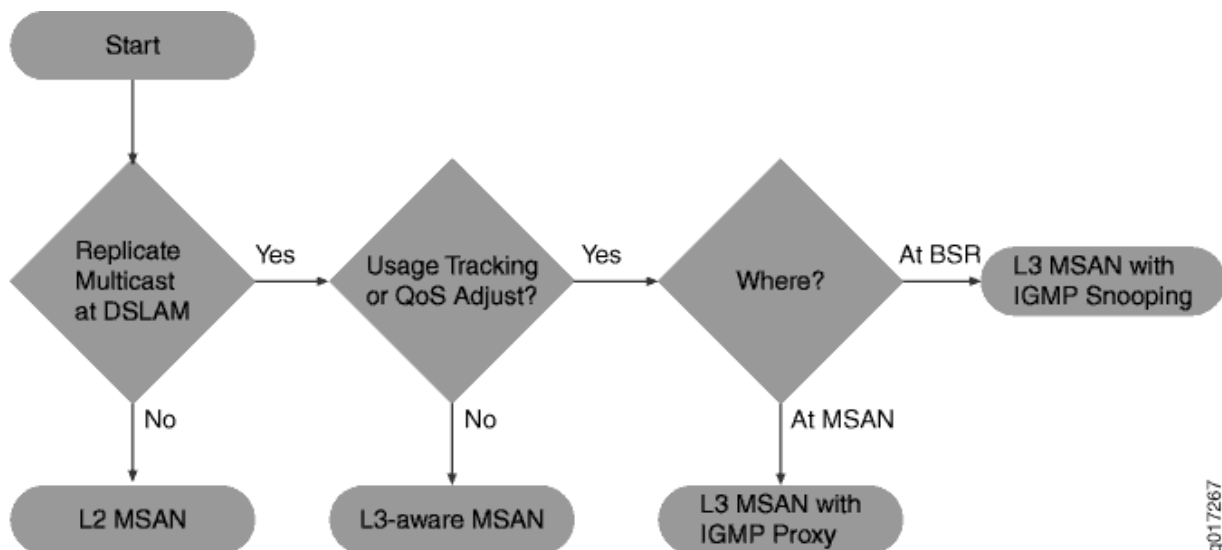
that acts as the centralized control point and prevents direct subscriber-to-subscriber communication. Ethernet Link Aggregation (LAG) provides the resiliency in this type of network.

Layer 2 DSLAMs cannot interpret IGMP, so they cannot selectively replicate IPTV channels.

- **Layer-3 aware MSAN**—This IP-aware MSAN can interpret and respond to IGMP requests by locally replicating a multicast stream and forwarding the stream to any subscriber requesting it. Layer 3 awareness is important when supporting IPTV traffic to perform channel changes (sometimes referred to as *channel zaps*). Static IP-aware MSANs always receive all multicast television channels. They do not have the ability to request that specific channels be forwarded to the DSLAM. Dynamic IP-aware DSLAMs, however, can inform the network to begin (or discontinue) sending individual channels to the DSLAM. Configuring IGMP proxy or IGMP snooping on the DSLAM accomplishes this function.
- **Layer-3 MSAN**—These MSANs use IP routing functionality rather than Layer 2 technologies to forward traffic. The advantage of this forwarding method is the ability to support multiple upstream links going to different upstream routers and improving network resiliency. However, to accomplish this level of resiliency, you must assign a separate IP subnetwork to each MSAN, adding a level of complexity that can be more difficult to maintain or manage.

In choosing a MSAN type, refer to [Figure 2 on page 4](#):

Figure 2: Choosing an MSAN Type



Ethernet MSAN Aggregation Options

IN THIS SECTION

- [Direct Connection | 5](#)
- [Ethernet Aggregation Switch Connection | 6](#)
- [Ring Aggregation Connection | 6](#)

Each MSAN can connect directly to an edge router (broadband services router or video services router), or an intermediate device (for example, an Ethernet switch) can aggregate MSAN traffic before being sent to the services router. [Table 1 on page 5](#) lists the possible MSAN aggregation methods and under what conditions they are used.

Table 1: Ethernet MSAN Aggregation Methods

Method	When Used
Direct connection	Each MSAN connects directly to the broadband services router and optional video services router.
Ethernet aggregation switch connection	Each MSAN connects directly to an intermediate Ethernet switch. The switch, in turn, connects to the broadband services router or optional video services router.
Ethernet ring aggregation connection	Each MSAN connects to a ring topology of MSANs. The head-end MSAN (the device closest to the upstream edge router) connects to the broadband services router.

You can use different aggregation methods in different portions of the network. You can also create multiple layers of traffic aggregation within the network. For example, an MSAN can connect to a central office terminal (COT), which, in turn, connects to an Ethernet aggregation switch, or you can create multiple levels of Ethernet aggregation switches prior to connecting to the edge router.

Direct Connection

In the direct connection method, each MSAN has a point-to-point connection to the broadband services router. If an intermediate central office exists, traffic from multiple MSANs can be combined onto a single connection using wave-division multiplexing (WDM). You can also connect the MSAN to a video

services router. However, this connection method requires that you use a Layer 3 MSAN that has the ability to determine which link to use when forwarding traffic.

When using the direct connection method, keep the following in mind:

- We recommend this approach when possible to simplify network management.
- Because multiple MSANs are used to connect to the services router, and Layer 3 MSANs generally require a higher equipment cost, this method is rarely used in a multiedge subscriber management model.
- Direct connection is typically used when most MSAN links are utilized less than 33 percent and there is little value in combining traffic from multiple MSANs.

Ethernet Aggregation Switch Connection

An Ethernet aggregation switch aggregates traffic from multiple downstream MSANs into a single connection to the services router (broadband services router or optional video services router).

When using the Ethernet aggregation switch connection method, keep the following in mind:

- Ethernet aggregation is typically used when most MSAN links are utilized over 33 percent or to aggregate traffic from lower speed MSANs (for example, 1 Gbps) to a higher speed connection to the services router (for example, 10 Gbps).
- You can use an MX Series router as an Ethernet aggregation switch. For information about configuring the MX Series router in Layer 2 scenarios, see the [Ethernet Networking User Guide for MX Series Routers](#).

Ring Aggregation Connection

In a ring topology, the remote MSAN that connects to subscribers is called the remote terminal (RT). This device can be located in the outside plant (OSP) or in a remote central office (CO). Traffic traverses the ring until it reaches the central office terminal (COT) at the head-end of the ring. The COT then connects directly to the services router (broadband services router or video services router).

NOTE: The RT and COT must support the same ring resiliency protocol.

You can use an MX Series router in an Ethernet ring aggregation topology. For information about configuring the MX Series router in Layer 2 scenarios, see the [Ethernet Networking User Guide for MX Series Routers](#).

LDP Pseudowire Autosensing Overview

IN THIS SECTION

- [Pseudowire Ingress Termination Background | 7](#)
- [Pseudowire Autosensing Approach | 8](#)
- [Sample Configuration | 9](#)

A pseudowire is a virtual link that is used to transport a Layer 2 service across an MPLS edge or access network. In a typical broadband edge or business edge network, one end of a pseudowire is terminated as a Layer 2 circuit on an access node, and the other end is terminated as a Layer 2 circuit on a service node that serves as either an aggregation node or an MPLS core network. Traditionally, both endpoints are provisioned manually through configuration. LDP pseudowire autosensing introduces a new provisioning model that allows pseudowire endpoints to be automatically provisioned and deprovisioned on service nodes based on LDP signaling messages. This model can facilitate the provisioning of pseudowires on a large scale. An access node uses LDP to signal both pseudowire identity and attributes to a service node. The identity is authenticated by a RADIUS server, and then used together with the attributes signaled by LDP and the attributes passed down by the RADIUS server to create the pseudowire endpoint configuration, including the Layer 2 circuit.

Pseudowire Ingress Termination Background

In a seamless MPLS-enabled broadband access or business edge network, Ethernet pseudowires are commonly used as virtual interfaces to connect access nodes to service nodes. Each pseudowire carries the bidirectional traffic of one or multiple broadband subscribers or business edge customers between an access node and a service node pair. The establishment of the pseudowire is usually initiated by the access node, based on either static configuration or dynamic detection of a new broadband subscriber or business edge customer arriving on a client-facing port on the access node.

Ideally, the access node should create one pseudowire per client port, where all subscribers or customers hosted by the port are mapped to the pseudowire. The alternative is where there is one pseudowire per client port (S-VLAN), and all subscribers or customers sharing a common S-VLAN on the port are mapped to the pseudowire. In either case, the pseudowire is signaled in the raw mode.

The S-VLAN, if not used to delimit service on the service node or combined with C-VLAN to distinguish subscribers or customers, will be stripped off before the traffic is encapsulated in pseudowire payload and transported to the service node. Individual subscribers or customers may be distinguished by C-VLAN, or a Layer 2 header such as DHCP and PPP, which will be carried in pseudowire payload to the service node. On the service node, the pseudowire is terminated. Individual subscribers or customers are

then demultiplexed and modeled as broadband subscriber interfaces, business edge interfaces (for example, PPPoE), Ethernet interfaces, or IP interfaces. Ethernet and IP interfaces may be further attached to service instances, such as VPLS and Layer 3 VPN instances.

In Junos OS, pseudowire ingress termination on service nodes is supported through the use of pseudowire service physical and logical interfaces. This approach is considered as superior in scalability to the old logical tunnel interface based approach, due to its capability of multiplexing and demultiplexing subscribers or customers over a single pseudowire. For each pseudowire, a pseudowire service physical interface is created on a selected Packet Forwarding Engine, which is called an anchor Packet Forwarding Engine. On top of this pseudowire service physical interface, a ps.0 logical interface (transport logical interface) is created, and a Layer 2 circuit or Layer 2 VPN is created to host the ps.0 logical interface as an attachment interface.

The Layer 2 circuit or Layer 2 VPN enables pseudowire signaling towards the access node, and the ps.0 logical interface serves the role of customer edge facing interface for the pseudowire. Further, one or multiple ps.n logical interfaces (also known as service logical interfaces, where $n > 0$) may be created on the pseudowire service physical interface to model individual subscriber/customer flows as logical interfaces. These interfaces can then be attached to desired broadband and business edge services or Layer 2 or Layer 3 VPN instances.

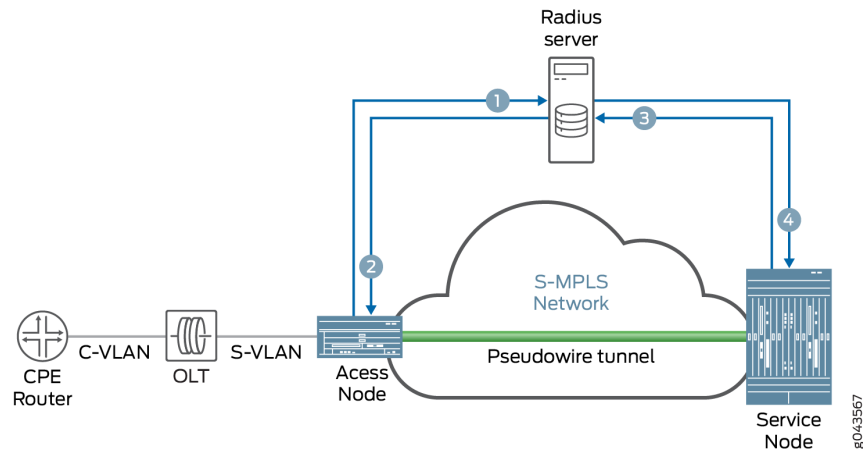
NOTE: Note that the purpose of the anchor Packet Forwarding Engine is to designate the Packet Forwarding Engine to process the bidirectional traffic of the pseudowire, including encapsulation, decapsulation, VLAN mux or demux, QoS, policing, shaping, and many more.

For Junos OS Release 16.2 and earlier, the creation and deletion of the pseudowire service physical interfaces, pseudowire service logical interfaces, Layer 2 circuits, and Layer 2 VPNs for pseudowire ingress termination rely on static configuration. This is not considered as the best option from the perspective of scalability, efficiency, and flexibility, especially in a network where each service node may potentially host a large number of pseudowires. The objective is to help service providers come out of static configuration in provisioning and deprovisioning pseudowire ingress termination on service nodes.

Pseudowire Autosensing Approach

In the pseudowire autosensing approach, a service node uses the LDP label mapping message received from an access node as a trigger to dynamically generate configuration for a pseudowire service physical interface, a pseudowire service logical interface, a Layer 2 circuit. Likewise, it uses the LDP label withdraw message received from the access node and LDP session down event as triggers to remove the generated configuration. In pseudowire autosensing, it is assumed that access nodes are the initiators of pseudowire signaling, and service nodes are the targets. In a network where a service may be hosted by multiple service nodes for redundancy or load balancing, this also provides access nodes with a select-and-connect model for service establishment. The basic control flow of pseudowire autosensing is shown in [Figure 3 on page 9](#)

Figure 3: Basic Control Flow of Pseudowire Autosensing



The basic control flow procedure of pseudowire autosensing is as follows:

1. Customer premises equipment (CPE) comes online and sends an Ethernet frame with C-VLAN to the optical line terminator (OLT). OLT adds S-VLAN to the frame and sends the frame to the access node. The access node checks with the RADIUS server to authorize the VLANs.
2. The RADIUS server sends an access accept to the access node. The access node creates a Layer 2 circuit and signals a pseudowire to the service node through an LDP label mapping message.
3. The service node accepts the label mapping message, and sends an access request with pseudowire information to the RADIUS server for authorization and for selection of a pseudowire service physical interface or a logical interface.
4. The RADIUS server sends an access accept to the service node with a service string specifying the selected pseudowire service physical interface or logical interface. The service node creates a Layer 2 circuit configuration, the pseudowire information, and the pseudowire service physical interface or logical interface. The service node signals the pseudowire towards the access node through an LDP label mapping message. The pseudowire comes up bidirectionally.

Sample Configuration

The following configuration explicitly marks the Layer 2 circuit as generated by autosensing. The pseudowire service physical interface and pseudowire service logical interface configuration are optional, depending on whether they preexist.

Router 0

```
[edit]
protocols {
```

```

Layer 2 circuit {
  neighbor 192.0.2.2 {
    interface ps0.0 {
      virtual-circuit-id 100;
      control-word;
      mtu 9100;
      auto-sensed;
    }
  }
}

```

Layer 2 Services on Pseudowire Service Interface Overview

IN THIS SECTION

- [Traffic from Customer LAN to MPLS | 11](#)
- [Traffic from Service Edge to Customer LAN | 12](#)
- [Pseudowire Service Interfaces | 13](#)
- [Sample Configuration | 14](#)

The pseudowire service logical interface supports the transport logical interface (psn.0) on the MPLS access side and service logical interfaces (psn.1 to psn.n) on the MPLS core side of the subscriber management network.

The pseudowire service on service logical interfaces psn.1 to psn.n are configured as Layer 2 interfaces in the bridge domain or in a virtual private LAN service (VPLS) instance. There is Layer 2 circuit or the Layer 2 VPN across MPLS access between an Ethernet aggregation device and a service edge device with the pseudowire service on transport logical interface psn.0 as the terminating interface of the Layer 2 circuit or the Layer 2 VPN at the service edge device.

Junos OS supports the pseudowire service on service logical interfaces psn.1 to psn.n in the bridge domain or VPLS instance, which receives traffic egressing from the pseudowire service on the transport logical interface at the service edge device. It also enables Layer 2 ingress features such as MAC learning, VLAN manipulations, and destination MAC look up on the pseudowire service on service logical interfaces.

When the traffic is in reverse direction, the destination MAC enters the Layer 2 domain at the service edge device, which is learned as the source MAC on the pseudowire service on service logical interfaces. Starting in Junos OS Release 17.1R1, the pseudowire logical tunnel interfaces support Ethernet VPLS, Ethernet bridge, VLAN VPLS, and VLAN bridge encapsulation next hops to exit Layer 2 traffic. Starting in Junos OS Release 18.4R1, the Layer 2 service support with the pseudowire service logical interfaces is extended to pseudowire service interfaces anchored over redundant logical tunnel interfaces as well. These Layer 2 services are supported only on pseudowire service on service logical interfaces (psn.1 to psn.n) and not on transport logical interface (psn.0). The Layer 2 output features such as VLAN manipulations and others are enabled on the pseudowire service interfaces. The traffic sent out of the interfaces enter the pseudowire service on transport logical interfaces which is the Layer 2 circuit interface between Ethernet aggregation and service edge devices across the MPLS access domain.

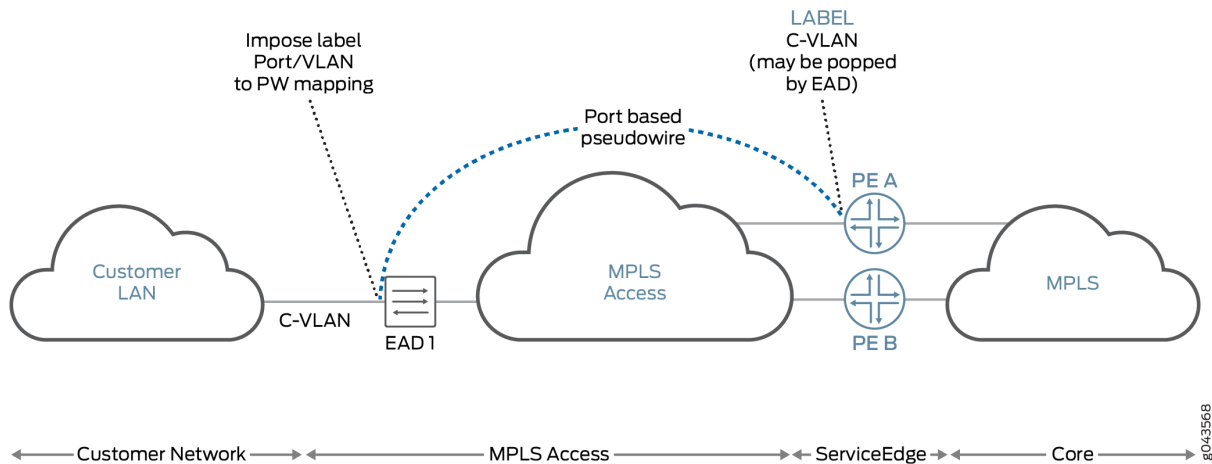
NOTE: For Junos OS Release 16.2 and earlier, Layer 2 encapsulations or features could not be configured on pseudowire service on service logical interfaces.

Traffic from Customer LAN to MPLS

VPLS-x and VPLS-y instances are configured on the MPLS core side of the service edge device (PE A). A Layer 2 circuit or Layer 2 VPN is configured between the Ethernet aggregation device (EAD 1) and the service edge device. ps0.0 (transport logical interface) is the local interface in the Layer 2 circuit or the Layer 2 VPN at PE A. Junos OS supports pseudowire service on service logical interface ps0.x (x>0) in VPLS instance VPLS-x (VLAN ID in VPLS-x = m) and pseudowire service on service logical interface ps0.y(y>0) in VPLS instance VPLS-y (VLAN ID in VPLS-y = n).

In [Figure 4 on page 12](#), when the traffic comes from EAD 1 to PE A (on either Layer 2 circuit or Layer 2 VPN) with any VLAN ID, the traffic will exit through ps0.0. Based on the VLAN ID in the traffic the pseudowire service on service logical interface is selected. For example, if VLAN ID is m, then the traffic will enter ps0.x and if VLAN ID is n, then the traffic will enter ps0.y.

Figure 4: Layer 2 Services for Pseudowire Service on Service Logical Interface



When traffic enters pseudowire service on the service logical interface ps0.n, where $n > 0$, the following steps are performed.

1. The source MAC learning should occur on the Layer 2 pseudowire service on the service logical interface. The source Packet Forwarding Engine for this MAC is the Packet Forwarding Engine of the logical tunnel interface on which the pseudowire service is anchored in a VPLS instance or bridge domain in the PE A device.
2. The destination MAC lookup is done at the entry side as an input bridge family feature list of pseudowire services on service logical interfaces.
 - If destination MAC lookup is successful, then the traffic is sent as unicast; otherwise, the destination MAC, broadcast MAC, and multicast MAC are flooded.
 - If destination MAC lookup fails for the traffic coming on a pseudowire service on a service logical interface, the `mIp` query command is sent to the Routing Engine and the other Packet Forwarding Engine in bridge domain or VPLS instance.
3. If a new MAC is learned on a pseudowire service on a service logical interface, then the `mIp add` command is sent to the Routing Engine and the other Packet Forwarding Engine in bridge domain or VPLS instance.

Traffic from Service Edge to Customer LAN

When traffic enters the VPLS instance or bridge domain at the service edge device and if the destination MAC in the traffic is learned on a pseudowire service on a service logical interface, then the token associated with that pseudowire service logical interface is set at the entry side. The traffic is then sent to the Packet Forwarding Engine on which the logical tunnel interface of the pseudowire service physical interface is anchored through a fabric. When this token is launched, it supports VLAN VPLS,

VLAN bridge, Ethernet VPLS, and Ethernet bridge encapsulations. The encapsulation next hop points to the egress logical interface feature list of the pseudowire service on the service logical interface to execute all the Layer 2 output features and send the packet to the entry side of the pseudowire service on transport logical interface ps0.0.

If the MAC query reaches the Packet Forwarding Engine on which the pseudowire service is anchored, then the Packet Forwarding Engine sends the response only when the MAC learned on the pseudowire service on the service logical interface is present. The Layer 2 token associated with the pseudowire service on the service logical interface seen after destination MAC lookup for the MAC learned on the pseudowire service on service logical interface should point to the next hop associated with the access side of the pseudowire service on service the logical interface.

The pseudowire service on the transport logical interface is the local interface ps0.0 of the Layer 2 circuit or Layer 2 VPN between the service edge and the Ethernet aggregation devices. Traffic is sent to the Ethernet aggregation device though the Layer 2 circuit or Layer 2 VPN across the MPLS access domain.

If the destination MAC traffic coming from the entry and exit side of the service edge device is unknown or multicast or broadcast, the traffic needs to be flooded. This requires an customer edge device flood next hop to include the pseudowire service on service logical interface, which acts as an access logical interface for the VPLS instance or bridge domain.

Pseudowire Service Interfaces

The following features are supported on pseudowire service interfaces:

- A pseudowire service interface is hosted over a logical tunnel interface (lt-x/y/z). The traffic from a transport pseudowire service on a logical interface to a subscriber pseudowire service on a logical interface is based on the available VLAN ID.
- Transfer of traffic from a subscriber pseudowire service on a logical interface to a transport pseudowire service on a logical interface is based on the channelID through an available loopback IP address.
- Pseudowire service on service logical interfaces are supported on the virtual routing and forwarding (VRF) routing instance.
- Pseudowire subscriber (ps) service on a trunk interface to terminate Layer 2 circuit instance in a VPLS-enabled virtual switch. The same Layer 2 circuit can also be terminated in the VPLS instance-type routing instance with different service logical interfaces and Layer 3 VPN VRF instance-type routing instance using another service logical interface as well.

Sample Configuration

The following sample configurations show a pseudowire service on a transport logical interface on a Layer 2 circuit, a pseudowire service on service logical interfaces in a bridge domain and a VPLS instance in a service edge device, and a pseudowire service on a trunk service interface in a VPLS instance:

Pseudowire service on a service logical interface in bridge domain on router 0

```
[edit]
  interfaces {
    ps0 {
      unit 0 {
        encapsulation ethernet-ccc;
      }
      unit 1 {
        encapsulation vlan-bridge;
        vlan-id 1;
      }
      unit 2 {
        encapsulation vlan-bridge;
        vlan-id 2;
      }
    }
    ge-0/0/0 {
      unit 1 {
        encapsulation vlan-bridge;
        vlan-id 1;
      }
      unit 2 {
        encapsulation vlan-bridge;
        vlan-id 2;
      }
    }
    ge-2/0/6 {
      unit 0 {
        family inet {
          address 10.11.2.1/24;
        }
        family mpls;
      }
    }
  }
}
```

```

protocols {
  mpls {
    label-switched-path to_192.0.2.2 {
      to 192.0.2.2;
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 192.0.3.3;
    }
  }
  l2-circuit {
    neighbor 192.0.2.2 {
      interface ps0.0 {
        virtual-circuit-id 100;
      }
    }
  }
}
bridge-domains {
  bd1 {
    domain-type bridge;
    vlan-id 1;
    interface ps0.1;
    interface ge-0/0/0.1;
  }
  bd2 {
    domain-type bridge;
    vlan-id 2;
    interface ps0.2;
    interface ge-0/0/0.2;
  }
}

```

Pseudowire service on a service logical interface in a VPLS instance on router 0

```

[edit]
interfaces {
  ps0 {
    unit 0 {
      encapsulation ethernet-ccc;
    }
  }
}

```

```
    }
    unit 1 {
        encapsulation vlan-vpls;
        vlan-id 1;
        family vpls;
    }
    unit 2 {
        encapsulation vlan-vpls;
        vlan-id 2;
        family vpls;
    }
}
ge-0/0/0 {
    unit 1 {
        encapsulation vlan-vpls;
        vlan-id 1;
        family vpls;
    }
    unit 2 {
        encapsulation vlan-vpls;
        vlan-id 2;
        family vpls;
    }
}
ge-2/0/6 {
    unit 0 {
        family inet {
            address 10.11.2.1/24;
        }
        family mpls;
    }
}
}
protocols {
    mpls {
        label-switched-path to_192.0.2.2 {
            to 192.0.2.2;
        }
    }
}
bgp {
    group RR {
        type internal;
        local-address 192.0.3.3;
    }
}
```

```

    }
  }
  l2-circuit {
    neighbor 192.0.2.2 {
      interface ps0.0 {
        virtual-circuit-id 100;
      }
    }
  }
}
routing-instances {
  vpls-1 {
    instance-type vpls;
    vlan-id 1;
    interface ps0.1;
    interface ge-0/0/0.1;
  }
  vpls-2 {
    instance-type vpls;
    vlan-id 2;
    interface ps0.2;
    interface ge-0/0/0.2;
  }
}
}

```

Pseudowire service on a trunk service interface in a VPLS instance on router 0

```

[edit]
  interfaces {
    ps0 {
      flexible-vlan-tagging;
      encapsulation flexible-ethernet-services;
      unit 0 {
        encapsulation ethernet-ccc;
      }
      unit 1 {
        family bridge {
          interface-mode trunk;
          vlan-id 1;
        }
      }
    }
  }
  ge-0/0/0 {

```

```

        unit 1 {
            encapsulation vlan-bridge;
            vlan-id 1;
            family bridge;
        }
    }
}
routing-instances {
    vpls-1 {
        instance-type virtual-switch;
        protocols {
            vpls {
                site PE3 {
                    interface ps0.1;
                    site-identifier 1;
                }
            }
        }
        bridge-domains {
            bd1 {
                vlan-id 1;
            }
        }
        interface ps0.1;
        route-distinguisher 65001:1;
        vrf-target target:1:1;
    }
}
}

```

Pseudowire service on a service logical interface in a Layer 2 circuit on router 0

```

[edit]
interfaces {
    ps0 {
        unit 0 {
            encapsulation ethernet-ccc;
        }
        unit 1 {
            encapsulation vlan-ccc;
            vlan-id 1;
        }
        unit 2 {

```

```
        encapsulation vlan-ccc;
            vlan-id 2;
        }
    }
    ge-0/0/0 {
        unit 1 {
            encapsulation vlan-vpls;
            vlan-id 1;
            family vpls;
        }
        unit 2 {
            encapsulation vlan-vpls;
            vlan-id 2;
            family vpls;
        }
    }
    ge-2/0/6 {
        unit 0 {
            family inet {
                address 10.11.2.1/24;
            }
            family mpls;
        }
    }
}
protocols {
    mpls {
        label-switched-path to_192.0.2.2 {
            to 192.0.2.2;
        }
    }
    bgp {
        group RR {
            type internal;
            local-address 192.0.3.3;
        }
    }
    l2-circuit {
        neighbor 192.0.2.2 {
            interface ps0.0 {
                virtual-circuit-id 100;
            }
        }
    }
}
```

```
neighbor 10.10.10.10 {
    interface ps0.1 {
        virtual-circuit-id 1;
    }
}
neighbor 10.11.11.11 {
    interface ps0.2 {
        virtual-circuit-id 2;
    }
}
}
```

Broadband Access Service Delivery Options

IN THIS SECTION

- [Digital Subscriber Line | 20](#)
- [Active Ethernet | 21](#)
- [Passive Optical Networking | 21](#)
- [Hybrid Fiber Coaxial | 22](#)

Four primary delivery options exist today for delivering broadband network service. These options include the following:

Digital Subscriber Line

Digital subscriber line (DSL) is the most widely deployed broadband technology worldwide. This delivery option uses existing telephone lines to send broadband information on a different frequency than is used for the existing voice service. Many generations of DSL are used for residential service, including Very High Speed Digital Subscriber Line 2 (VDSL2) and versions of Asymmetric Digital Subscriber Line (ADSL, ADSL2, and ADSL2+). These variations of DSL primarily offer asymmetric residential broadband service where different upstream and downstream speeds are implemented. (VDSL2 also supports symmetric operation.) Other DSL variations, like High bit rate Digital Subscriber Line (HDSL) and

Symmetric Digital Subscriber Line (SDSL), provide symmetric speeds and are typically used in business applications.

The head-end to a DSL system is the Digital Subscriber Line Access Multiplexer (DSLAM). The demarcation device at the customer premise is a DSL modem. DSL service models are defined by the Broadband Forum (formerly called the DSL Forum).

Active Ethernet

Active Ethernet uses traditional Ethernet technology to deliver broadband service across a fiber-optic network. Active Ethernet does not provide a separate channel for existing voice service, so VoIP (or TDM-to-VoIP) equipment is required. In addition, sending full-speed (10 or 100 Mbps) Ethernet requires significant power, necessitating distribution to Ethernet switches and optical repeaters located in cabinets outside of the central office. Due to these restrictions, early Active Ethernet deployments typically appear in densely populated areas.

Passive Optical Networking

Passive Optical Networking (PON), like Active Ethernet, uses fiber-optic cable to deliver services to the premises. This delivery option provides higher speeds than DSL but lower speeds than Active Ethernet. Though PON provides higher speed to each subscriber, it requires a higher investment in cable and connectivity.

A key advantage of PON is that it does not require any powered equipment outside of the central office. Each fiber leaving the central office is split using a non-powered optical splitter. The split fiber then follows a point-to-point connection to each subscriber.

PON technologies fall into three general categories:

- ATM PON (APON), Broadband PON (BPON), and Gigabit-capable PON (GPON)—PON standards that use the following different delivery options:
 - APON—The first passive optical network standard is primarily used for business applications.
 - BPON—Based on APON, BPON adds wave division multiplexing (WDM), dynamic and higher upstream bandwidth allocation, and a standard management interface to enable mixed-vendor networks.
 - GPON—GPON is based on BPON but supports higher rates, enhanced security, and a choice of which Layer 2 protocol to use (ATM, Generic Equipment Model [GEM], or Ethernet).
- Ethernet PON (EPON)—Provides capabilities similar to GPON, BPON, and APON, but uses Ethernet standards. These standards are defined by the IEEE. Gigabit Ethernet PON (GEAPON) is the highest speed version.

- Wave Division Multiplexing PON (WDM-PON)—A nonstandard PON which, as the name implies, provides a separate wavelength to each subscriber.

The head-end to a PON system is an Optical Line Terminator (OLT). The demarcation device at the customer premises is an Optical Network Terminator (ONT). The ONT provides subscriber-side ports for connecting Ethernet (RJ-45), telephone wires (RJ-11) or coaxial cable (F-connector).

Hybrid Fiber Coaxial

Multi-System Operators (MSOs; also known as *cable TV operators*) offer broadband service through their hybrid fiber-coaxial (HFC) network. The HFC network combines optical fiber and coaxial cable to deliver service directly to the customer. Services leave the central office (CO) using a fiber-optic cable. The service is then converted outside of the CO to a coaxial cable *tree* using a series of optical nodes and, where necessary, through a trunk radio frequency (RF) amplifier. The coaxial cables then connect to multiple subscribers. The demarcation device is a cable modem or set-top box, which talks to a Cable Modem Termination System (CMTS) at the MSO *head-end* or primary facility that receives television signals for processing and distribution. Broadband traffic is carried using the Data Over Cable Service Interface Specification (DOCSIS) standard defined by CableLabs and many contributing companies.

Broadband Delivery and FTTx

Many implementations use existing copper cabling to deliver signal to the premises, but fiber-optic cable connectivity is making its way closer to the subscriber. Most networks use a combination of both copper and fiber-optic cabling. The term *fiber to the x* (FTTx) describes how far into the network fiber-optic cabling runs before a switch to copper cabling takes place. Both PON and Active Ethernet can use fiber-optic portion of the network, while xDSL is typically used on the copper portion. This means that a single fiber-optic strand may support multiple copper-based subscribers.

Increasing the use of fiber in the network increases cost but it also increases network access speed to each subscriber.

The following terms are used to describe the termination point of fiber-optic cable in a network:

- Fiber to the Premises (FTTP), Fiber to the Home (FTTH), Fiber to the Business (FTTB)—Fiber extends all the way to the subscriber. PON is most common for residential access, although Active Ethernet can be efficiently used in dense areas such as apartment complexes. Active Ethernet is more common for delivering services to businesses.
- Fiber to the Curb (FTTC)—Fiber extends most of the way (typically, 500 feet/150 meters or less) to the subscriber. Existing copper is used for the remaining distance to the subscriber.
- Fiber to the Node/Neighborhood (FTTN)—Fiber extends to within a few thousand feet of the subscriber and converted to xDSL for the remaining distance to the subscriber.

- Fiber to the Exchange (FTTE)—A typical central office-based xDSL implementation in which fiber is used to deliver traffic to the central office and xDSL is used on the existing local loop.

Understanding BNG Support for Cascading DSLAM Deployments Over Bonded DSL Channels

IN THIS SECTION

- [Benefits of Cascading DSLAM Deployments Over Bonded DSL Channels | 24](#)
- [4-Level Scheduler Hierarchy | 24](#)
- [Use Cases of Cascading DSLAM Deployments Over Bonded DSL Channels | 25](#)
- [Bonded DSL for Copper-To-The-Building \(CuTTB\) | 25](#)
- [Hybrid PON + G.fast | 26](#)
- [Supported Features | 26](#)

Junos OS supports configuring and maintaining the access lines between access nodes and their ANCP subscribers using DSL access multiplexer as the broadband access technology for Copper-to-the-Building (CuTTB) and Fiber-to-the-Building (FTTB). When multiple subscribers share the same access line, the access line could be one of the following types:

- PON, Fiber-to-the-Building (FTTB)
- Bonded DSL Copper-To-The-Building (CTTB)

Starting in Junos OS Release 18.2R1, Passive Optical Network (PON) access technologies are supported with four levels of quality-of-service (QoS) scheduler hierarchy for residential subscribers in a BBE deployment. This feature extends the Access Node Control Protocol (ANCP) implementation to handle network configuration for residential customers that use PON as the broadband access technology for both CuTTB and FTTB. ANCP uses a statically controlled traffic-control profile on the interface-set for shaping at the subscriber level at the intermediate node to which the subscribers are connected. New DSL types are provided to support access line rate adjustment for the new access technologies.

A new RADIUS VSA, Inner-Tag-Protocol-Id 26-211 is introduced to fetch the inner VLAN Tag Protocol Identifier value for L2BSA subscribers to enable maintaining one dynamic profile instead of two separate dynamic profiles. A new Junos OS dynamic profile variable `$junos-inner-vlan-tag-protocol-id` allows a VLAN map's inner-tag-protocol-id to be set by RADIUS or a predefined default value provided in the configuration.

Benefits of Cascading DSLAM Deployments Over Bonded DSL Channels

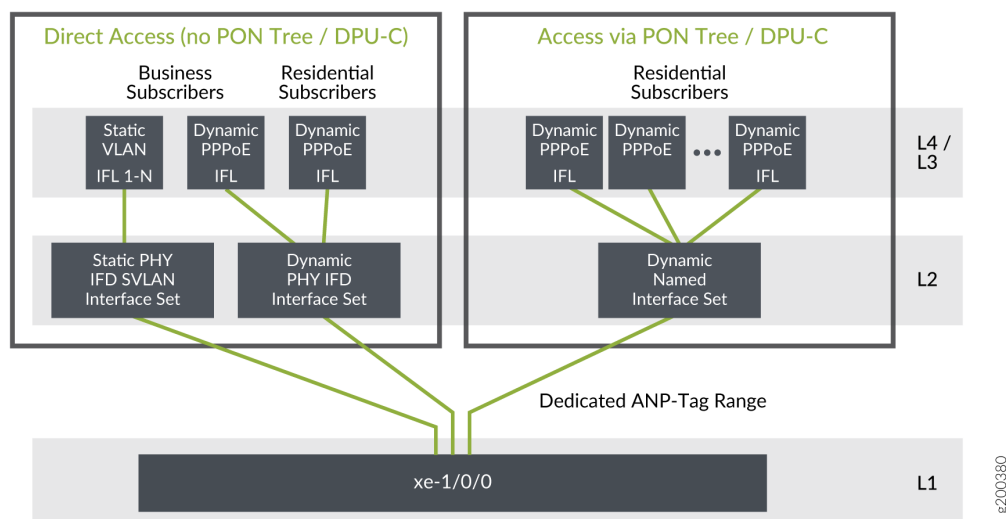
This feature is useful to support access network deployments where multiple subscribers share the same access line aggregated by an intermediate node between the access node and the home routing gateways. Another benefit is to conserve Layer 2 CoS nodes. Typically a dummy Layer 2 node is created for each residential household, which could exhaust Layer 2 CoS resources. Therefore, network models using bonded DSL, G.Fast, and PON access models can conserve Layer 2 CoS nodes.

4-Level Scheduler Hierarchy

Junos OS supports 4-Level QoS scheduler hierarchy minimally supporting residential and L2BSA access over Copper-to-the-Building (CTTB) or Fiber-to-the-Building access network deployments. The following QoS scheduler hierarchy levels are supported:

- Level 1 Port (Physical interface or AE)
- Level 2 Access Line (Logical interface set, represents a collection of subscribers sharing a given access line aggregated by an intermediate node)
- Level 3 Subscriber sessions
- Level 4 Queues (services)

Figure 5: Scheduler Hierarchy



In [Figure 5 on page 24](#), residential and L2BSA access require only 4-level scheduler hierarchy. Business subscriber access is currently not supported and hence 4-level scheduler hierarchy is sufficient for CuTTB and PON services targeted to an apartment building.

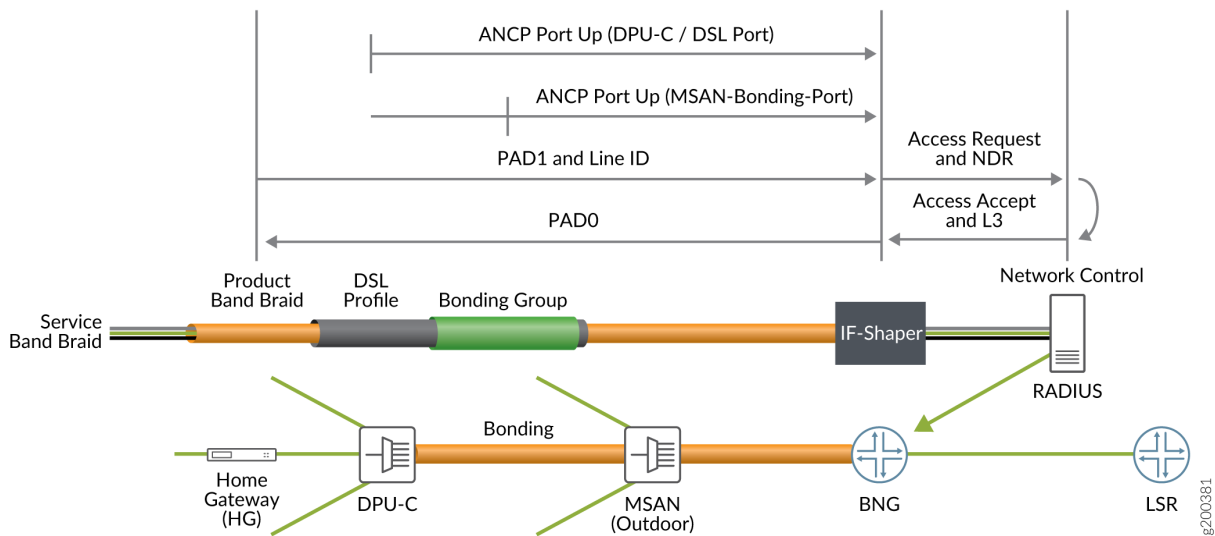
Use Cases of Cascading DSLAM Deployments Over Bonded DSL Channels

Bonded DSL for copper to the building (CuTTB) introduces an intermediate node Distribution Point Unit-Copper (DPU-C) between the DSL access multiplexer (DSLAM) and a cluster of subscribers at the customer location. Shared access line deployment models may be of type Passive-Optical-Network (PON) or bonded DSL copper lines. Example intermediate nodes are listed below:

- DPU-C - bonded DSL for Copper-To-The-Building (CTTB)
- ONU - PON (Fiber-to-the-Building (FTTB))
- Hybrid PON and G.Fast

Bonded DSL for Copper-To-The-Building (CuTTB)

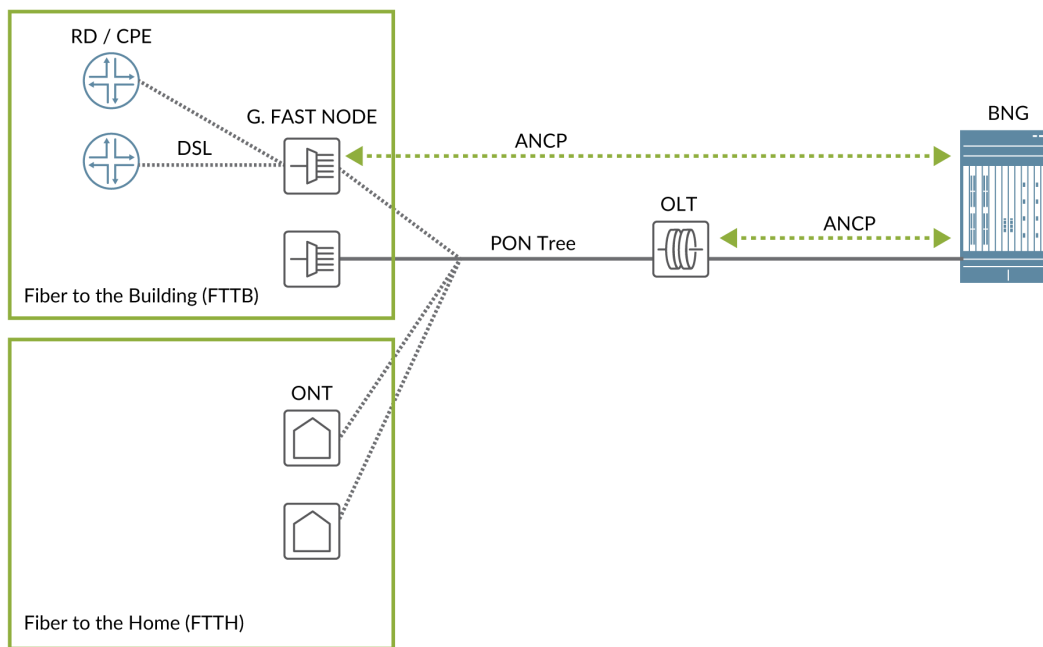
Figure 6: Bonded DSL/CuTTB



In [Figure 6 on page 25](#), each DPU-C has an ANCP session to report access line parameters of individual subscribers connected to the node. The MSAN also has an ANCP session to report access line parameters of the bonded DSL access line to the DPU-C. All subscribers connected to the DPU-C are thus subject to the DSL access-line downstream rate, the DPU-C subscribers are grouped together in an interface set. You can adjust the speeds reported in this Port-Up and apply to the CoS node for the corresponding interface set maintaining the semantics of the CoS adjustment control profile that is used for individual subscriber lines. The access model consists of a hybrid of bonded DSL access and conventional unbonded access. The DPU-C and the Multi Service Access Node (MSAN) ANCP sessions are completely independent and the PPPoE-IA tags only reflect the attributes reported in the dPU-C ANCP session

Hybrid PON + G.fast

Figure 7: Hybrid PON + G.fast



In [Figure 7 on page 26](#), the OLT has an ANCP session with the BNG and proxies for all downstream native PON nodes. G.fast DSL subscribers are connected to an intermediate node, which has a PON connection to the intermediate ONU in front of the OLT.

A hybrid access network connects DSL based subscriber lines using both PON access and G.fast nodes with an intermediate node between the OLT and home gateways (HGs). Both businesses and residences are connected to the intermediate node, which is the PON leaf. Shaping is required both at the subscriber level and at the PON leaf level. The G.fast subscribers are associated with the intermediate ONU like a native PON subscriber. New DSL type TLVs are supported by the AN and their values are reported in the ANCP Port-Up for the corresponding subscriber access line. However, it is still not possible to distinguish between an intermediate node and a conventional connection for a given PPPoE session.

Supported Features

- Support ANCP-based traffic shaping on dynamic ifsets.
- Preservation of PPPoE-IA and ANCP independence by CLI configuration for residential subscribers.

- New Juniper VSA, ERX-Inner-Vlan-Tag-Protocol-Id (4874-26-211) is supported to source the inner VLAN Tag Protocol Identifier value for L2BSA subscribers as an optimization to maintain two, separate dynamic profiles, one for TPID - 0x88a8 and one for 0x8100, and sourcing the desired value by returning 26-4874-174 (Client-profile-Name) in the Access-Accept.
- The following additional type values for the DSL type TLV are supported. All subscribers include these DSL type TLVs in the PPPoE PADR messages's PPPoE IA tags.
 - (8) G.fast
 - (9) VDSL2 Annex Q
 - (10) SDSL bonded
 - (11) VDSL2 bonded
 - (12) G,fast bonded
 - (13) VDSL2 Annex Q bonded

Detection of Backhaul Line Identifiers and Autogeneration of Intermediate Node Interface Sets

Before you begin, you must confirm that your existing access nodes or IAs are not already inserting strings that begin with the # character. Because this is a system-level configuration, parsing applies to all ANCP access nodes and PPPoE IAs globally. The leading # character is not configurable. Parsing is disabled by default in case some providers use that character for some other purpose.

Starting in Junos OS Release 18.4R1, you can configure the router to detect a logical intermediate node in an access network. The node identifies subscribers that are connected to the same shared media, such as a PON tree or a bonded copper line that connects to a DPU-C for CuTTB. When you configure this detection, the router parses the ANCP Access-Aggregation-Circuit-ID-ASCII attribute (TLV 0x03) that is received either in the ANCP Port Up message or PPPoE PADR IA tags. If the TLV string begins with the # character, the string is a backhaul line identifier that is unique across the network to identify the bonded DSL line or the PON tree. The same string is reported in the TLV or IA for all subscribers connected to that DPU-C or PON.

The portion of the string after the # character represents the logical intermediate node. It is used as the name of the dynamic interface set for the CoS Level 2 node that groups the subscribers using that intermediate node. This interface set is known as the parent interface set. Every PPPoE or VLAN (L2BSA) logical interface with the same value for TLV 0x03 is a member of that interface set.

NOTE: The TLV value must match the requirements for interface set naming; it can include alphanumeric characters and the following special characters:

% / = + - : ; @ . _

This portion of the string also sets the value of the `$junos-aggregation-interface-set-name` predefined variable in the dynamic profile. This value is used as the name of a CoS Level 2 interface set that groups the subscribers sharing that string. It overrides the predefined variable default, which uses the value of `$junos-phy-ifd-interface-set-name` as the name of the interface set.

For example, if the value of the TLV string is `#TEST-DPU-C-100`, the value of the predefined variable—and consequently the name of the interface set—becomes `TEST-DPU-C-100`.

NOTE: The Access-Loop-Remote-ID (TLV (0x02) is similarly parsed for the # character, but the resulting string is not used in the current release.

NOTE: Intermediate node detection is supported only for 4-level scheduler hierarchies, so business access is limited to conventional DSL access MPCs.

To enable parsing of the Access-Aggregation-Circuit-ID-ASCII TLV and setting the interface set name:

1. Specify detection of hierarchical access networks and extraction of the node string.

```
[edit system access-line]
user@host# set hierarchical-access-network-detection
```

2. Configure the dynamic profile to use the Access-Aggregation-Circuit-ID-ASCII string for the interface set name.

```
[edit dynamic-profile interfaces]
user@host# set interface-set $junos-aggregation-interface-set-name
```

The following sample configuration shows a dynamic profile for L2BSA subscribers. Three things to note here are the following:

- A default value of `$junos-phy-ifd-interface-set-name` is defined for the `$junos-aggregation-interface-set-name` predefined variable.

- The name of the interface set is configured to be the value of `$junos-aggregation-interface-set-name`.
- The CoS scheduler configuration specifies an interface named with the value of `$junos-aggregation-interface-set-name`.

When `hierarchical-access-network-detection` is configured for the access lines, then the name of the Level 2 scheduler interface set is determined as follows:

- When TLV 0x03 begins with #, then `$junos-aggregation-interface-set-name` is the remainder of the string, excluding the initial #.
- When TLV 0x03 begins with any other character, then `$junos-aggregation-interface-set-name` is the value of `$junos-phy-ifd-interface-set-name`.

```
[edit dynamic-profiles L2BSA-subscriber]
predefined-variable-defaults {
  aggregation-interface-set-name phy-ifd-interface-set-name;
  cos-shaping-rate 1g;
  cos-scheduler-map schedmap_L2BSA;
  inner-vlan-tag-protocol-id 0x88a8;
}
routing-instances {
  "$junos-routing-instance" {
    interface "$junos-interface-name";
  }
}
interfaces {
  interface-set $junos-aggregation-interface-set-name {
    interface "$junos-interface-ifd-name" {
      unit "$junos-interface-unit";
    }
  }
  "$junos-interface-ifd-name" {
    unit "$junos-interface-unit" {
      encapsulation vlan-vpls;
      no-traps;
      vlan-id "$junos-vlan-id";
      input-vlan-map {
        swap-push;
        inner-tag-protocol-id "$junos-inner-vlan-tag-protocol-id"
        vlan-id "$junos-vlan-map-id";
        inner-vlan-id "$junos-inner-vlan-map-id";
      }
    }
  }
}
```

```

    }
    output-vlan-map {
        pop-swap;
        inner-tag-protocol-id 0x8100;
    }
    family vpls;
}
}
}
}
class-of-service {
    traffic-control-profiles {
        L2BSAShaper {
            scheduler-map "$junos-cos-scheduler-map";
            shaping-rate "$junos-cos-shaping-rate" burst-size 17k;
            overhead-accounting frame-mode cell-mode-bytes 6;
        }
        L2iflsetShaper {
            shaping-rate 1G burst-size 17k;
        }
    }
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
            output-traffic-control-profile L2BSAShaper;
            classifiers {
                ieee-802.1 L2BSA vlan-tag outer;
            }
            rewrite-rules {
                ieee-802.1 L2BSA vlan-tag outer;
            }
        }
    }
}
interface-set "$junos-aggregation-interface-set-name" {
    output-traffic-control-profile L2iflsetShaper;
}
}
}
}

```

Release History Table

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, the Layer 2 service support with the pseudowire service logical interfaces is extended to pseudowire service interfaces anchored over redundant logical tunnel interfaces as well.
18.4R1	Starting in Junos OS Release 18.4R1, you can configure the router to detect a logical intermediate node in an access network.
17.1R1	Starting in Junos OS Release 17.1R1, the pseudowire logical tunnel interfaces support Ethernet VPLS, Ethernet bridge, VLAN VPLS, and VLAN bridge encapsulation next hops to exit Layer 2 traffic.

RELATED DOCUMENTATION

Subscriber Management Overview

[MPLS Pseudowire Subscriber Logical Interfaces | 334](#)

Juniper Networks VSAs Supported by the AAA Service Framework

High Availability for Subscriber Access Networks

IN THIS SECTION

- [Unified ISSU for High Availability in Subscriber Access Networks | 32](#)
- [Verifying and Monitoring Subscriber Management Unified ISSU State | 33](#)
- [Graceful Routing Engine Switchover for Subscriber Access Networks | 34](#)
- [Minimize Traffic Loss Due to Stale Route Removal After a Graceful Routing Engine Switchover | 35](#)

This topic is a high-level overview of high availability for DHCP, L2TP, and PPP access networks.

Unified ISSU for High Availability in Subscriber Access Networks

A unified in-service software upgrade (unified ISSU) enables you to upgrade between two different Junos OS Releases with no disruption on the control plane and with minimal disruption of traffic. The routers preserve the active subscriber sessions and session services across the upgrade, so that they continue after the upgrade has completed.

The unified ISSU feature supports the PPPoE, DHCP, and L2TP access models for subscriber management. Unified ISSU support for the DHCP and L2TP access models was added in Junos OS Release 14.1.

- For static and dynamic PPPoE access, unified ISSU supports the following:
 - Terminated, non-tunneled PPPoE connections configured with static or dynamic PPP logical interfaces and static or dynamic underlying interfaces
 - Subscriber services on single-link PPP interfaces
 - Preservation of statistics for accounting, filter, and CoS on MPC/MIC interfaces

NOTE: Unified ISSU for the subscriber management PPPoE access model *does not support* Multilink Point-to-Point Protocol (MLPPP) bundle interfaces. MLPPP bundle interfaces require the use of an Adaptive Services PIC or Multiservices PIC to provide PPP subscriber services. These PICs do not support unified ISSU.

- For DHCP access, unified ISSU supports the following:
 - DHCPv4 local server, DHCPv4 relay, DHCPv6 local server, DHCPv6 relay, and DHCP relay proxy
 - Preservation of accounting, filter, and class-of-service (CoS) statistics for DHCP subscribers on MPC/MIC interfaces on MX Series routers
- For L2TP access, unified ISSU supports both the LAC and the LNS. When an upgrade is initiated, the LAC completes any L2TP negotiations that are in progress but rejects any new negotiations until the upgrade has completed. No new tunnels or sessions are established during the upgrade. Subscriber logouts are recorded during the upgrade and are completed after the upgrade has completed.

See [Getting Started with Unified In-Service Software Upgrade](#) for a description of the supported platforms and modules, CLI statements, and procedures you use to configure and initiate unified ISSU. You can use the `issu` flag with the `traceoptions` statement to trace subscriber management unified ISSU events. You can also use the `show system subscriber-management summary` command to display information about the unified ISSU state.

Verifying and Monitoring Subscriber Management Unified ISSU State

IN THIS SECTION

- Purpose | 33
- Action | 33

Purpose

Display the state of unified ISSU for subscriber management features.

Action

The first example indicates that control plane quiescing as part of unified ISSU is not in progress (for example, unified ISSU has not been started, has already completed, or control plane quiescing has not started). The second example shows that unified ISSU is in progress and that a participating subscriber management daemon requires 198 seconds to quiesce the control plane.

```
user@host> show system subscriber-management summary
```

```
General:
```

Graceful Restart	Enabled
Mastership	Master
Database	Available
Chassisd ISSU State	IDLE
ISSU State	IDLE
ISSU Wait	0

```
user@host> show system subscriber-management summary
```

```
General:
```

Graceful Restart	Enabled
Mastership	Master
Database	Available
Chassisd ISSU State	DAEMON_ISSU_PREPARE
ISSU State	PREPARE
ISSU Wait	198

Graceful Routing Engine Switchover for Subscriber Access Networks

IN THIS SECTION

- DHCP | 34
- L2TP | 34

The *graceful Routing Engine switchover* (GRES) feature in Junos OS enables a router with redundant Routing Engines to continue forwarding packets, even if one Routing Engine fails. GRES preserves interface and kernel information. Traffic is not interrupted. However, GRES does not preserve the control plane.

To enable GRES support on MX Series routers, include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level.

DHCP

For MX Series routers, the extended DHCP local server and the DHCP relay agent applications both maintain the state of active DHCP client leases in the session database. The extended DHCP application can recover this state if the DHCP process fails or is manually restarted, thus preventing the loss of active DHCP clients in either of these circumstances. However, the state of active DHCP client leases is lost if a power failure occurs or if the kernel stops operating (for example, when the router is reloaded) on a single Routing Engine.

You cannot disable graceful Routing Engine switchover support for the extended DHCP application when the router is configured to support graceful Routing Engine switchover.

For more information about using graceful Routing Engine switchover, see [Understanding Graceful Routing Engine Switchover](#).

L2TP

GRES is supported on MX Series routers acting as either the L2TP LAC or LNS. In the event that L2TP (`jl2tpd`, the L2TP universal edge process) restarts or that the router fails over from the active routing engine (RE) to the standby RE, L2TP GRES ensures that the following occurs:

- The LAC and the LNS recover destinations, tunnels, and sessions that were already established at the time of the failure or restart.

- The LAC and the LNS respond to tunnel keepalive requests received during the switchover for established tunnels, but do not generate any keepalives until the switchover is complete.
- The LAC and the LNS delete all the tunnels and sessions that are not in the Established state.
- The LAC and the LNS reject requests to create new tunnels and sessions.
- The LAC and the LNS send another disconnect notification to the peer for sessions and tunnels that are already in the Disconnecting state at the time of the failure or restart. For sessions and tunnels that were coming up at that time, the LAC and LNS send a disconnect notification to the peer.
- The LAC and the LNS restart timers for the full timeout period for recovered L2TP destinations, tunnels, and sessions.

If a graceful Routing Engine switchover (GRES) is triggered by an operational mode command, the state of aggregated services interfaces (ASIs) are not preserved. For example:

```
request interface <switchover | revert> asi-interface
```

However, if GRES is triggered by a CLI commit or FPC restart or crash, the backup Routing Engine updates the ASI state. For example:

```
set interface si-x/y/z disable
commit
```

Or:

```
request chassis fpc restart
```

Minimize Traffic Loss Due to Stale Route Removal After a Graceful Routing Engine Switchover

During a *graceful Routing Engine switchover* (GRES), access routes and access-internal routes for DHCP and PPP subscriber management can become stale. After the GRES, the router removes any such stale routes from the forwarding table. Some traffic is lost if the stale routes are removed before the routes are reinstalled.

In subscriber networks with graceful restart and routing protocols such as BGP and OSPF configured, the router purges any remaining stale access routes and access-internal routes as soon as the graceful

restart operation completes, which can occur very soon after completion of the graceful Routing Engine switchover.

In subscriber networks with *nonstop active routing* (NSR) and routing protocols such as BGP and OSPF configured, the routing protocol process (rpd) immediately purges the stale access routes and access-internal routes that correspond to subscriber routes.

You can reduce the risk of this traffic loss by configuring the router to delay the removal of stale routes after a GRES. The delay period is a nonconfigurable 180 seconds (3 minutes). The router retains the stale routes for the duration of the period, which is long enough for the DHCP client process (jdhcpd), PPP client process (jpppd), or routing protocol process (rpd) to reinstall the access routes and access-internal routes before the router removes the stale routes from the forwarding table. The risk of traffic loss is minimized because the router always has available subscriber routes for DHCP subscribers and PPP subscribers.

To configure the router to delay removal (flushing) of access-routes and access-internal routes after a graceful Routing Engine switchover:

1. Specify that you want to configure subscriber management.

```
[edit system services]
user@host# edit subscriber-management
```

2. Configure the router to wait 180 seconds before removing access-routes and access-internal routes after a graceful Routing Engine switchover.

```
[edit system services subscriber-management]
user@host# set gres-route-flush-delay
```

Release History Table

Release	Description
14.1	Unified ISSU support for the DHCP and L2TP access models was added in Junos OS Release 14.1.

RELATED DOCUMENTATION

[Getting Started with Unified In-Service Software Upgrade](#)

[Routes for DHCP and PPP Subscriber Access Networks | 37](#)

Routes for DHCP and PPP Subscriber Access Networks

IN THIS SECTION

- [Access and Access-Internal Routes for Subscriber Management | 37](#)
- [Configuring Dynamic Access Routes for Subscriber Management | 38](#)
- [Configuring Dynamic Access-Internal Routes for DHCP and PPP Subscribers | 40](#)
- [Suppressing DHCP Access, Access-Internal, and Destination Routes | 41](#)
- [Preventing DHCP from Installing Access, Access-Internal, and Destination Routes by Default | 42](#)
- [Verifying the Configuration of Access and Access-Internal Routes for DHCP and PPP Subscribers | 43](#)

Access and Access-Internal Routes for Subscriber Management

DHCP and PPP on the router use both access routes and access-internal routes to represent either the subscriber or the networks behind the attached router. An access route represents a network behind an attached router, and is set to a preference of 13. An access-internal route is a /32 route that represents a directly attached subscriber, and is set to a preference of 12.

Access routes typically are used to apply the values of the RADIUS Framed-Route attribute [22] for IPv4 routes and the Framed-IPv6-Route attribute [99] for IPv6 routes. A framed route consists of a prefix that represents a public network behind the CPE, a next-hop gateway, and optional route attributes consisting of a combination of metric, preference, and tag. The only mandatory component of the framed route is the prefix. The next-hop gateway can be specified explicitly in the framed route, as 0.0.0.0, ::0, or the subscriber's fixed address assigned by the Framed-IP-Address (8) or Framed-IPv6-Prefix (97) attribute (common practice for business subscribers). Alternatively, the absence of the gateway address implies address 0.0.0.0. The address 0.0.0.0 or ::0, whether implicit or explicitly configured, resolves to the subscriber's assigned address (host route). Consequently, the convention is that the next-hop gateway is the subscriber's IP address.

You can configure a dynamic profile to use predefined variables to dynamically configure access routes using the values specified in the RADIUS attribute. To configure access routes include the access stanza at the [edit dynamic-profiles *profile-name* routing-options] hierarchy level.

Starting in Junos OS Release 15.1, we recommend that you use only access routes for framed route support. We recommend that you do not use access-internal routes in the dynamic profile configuration.

If the RADIUS Framed-Route attribute [22] or Framed-IPv6-Route attribute [99] does not specify the next-hop gateway—as is common—the variable representing the next-hop, `$junos-framed-route-nexthop` or `$junos-framed-route-ipv6-nexthop`, automatically resolves to the subscriber's IP address. If you configure the `access-internal` statement in the dynamic profile, it is ignored.

NOTE: Starting in Junos OS Release 15.1R4, the router no longer supports a configuration where a static route points to a next hop that is tied to a subscriber. Typically, this might occur when RADIUS assigns the next hop with the Framed-IP-Address attribute. An alternative to this misconfiguration is to have the RADIUS server provide a Framed-Route attribute that matches the static route.

Configuring Dynamic Access Routes for Subscriber Management

You can dynamically configure access routes for DHCP and PPP subscribers based on the values specified in the following RADIUS attributes:

- For IPv4 access routes, use the variable, `$junos-framed-route-ip-address-prefix`. The route prefix variable is dynamically replaced with the value in Framed-Route RADIUS attribute [22].
- For IPv6 access routes, use the variable, `$junos-framed-route-ipv6-address-prefix`. The variable is dynamically replaced with the value in Framed-IPv6-Route RADIUS attribute [99].

To dynamically configure access routes:

1. Configure the route prefix for the access route as a variable.

For IPv4:

```
[edit dynamic-profiles profile-name routing-options]
user@host# edit access route $junos-framed-route-ip-address-prefix
```

For IPv6:

```
[edit dynamic-profiles profile-name routing-options]
user@host# edit access route $junos-framed-route-ipv6-address-prefix
```

2. Configure the next-hop address as a variable.

For IPv4:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set next-hop $junos-framed-route-nexthop
```

For IPv6:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ipv6-
address-prefix"]
user@host# set next-hop $junos-framed-route-ipv6-nexthop
```

3. Configure the metric as a variable.

For IPv4:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set metric $junos-framed-route-cost
```

For IPv6:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set metric $junos-framed-route-ipv6-cost
```

4. Configure the preference as a variable.

For IPv4:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set preference $junos-framed-route-distance
```

For IPv6:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set preference $junos-framed-route-ipv6-distance
```

5. Configure the tag as a variable.

IPv4:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set tag $junos-framed-route-tag
```

IPv6:

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix"]
user@host# set tag $junos-framed-route-ipv6-tag
```

Starting in Junos OS Release 15.1, we recommend that you use only access routes for framed route support. We recommend that you do not use access-internal routes. If the RADIUS Framed-Route attribute (22) or Framed-IPv6-Route attribute [99] does not specify the next-hop gateway—as is common—the variable representing the next-hop, `$junos-framed-route-nexthop`, is automatically resolved. If you configure the `access-internal` statement in the dynamic profile, it is ignored.

Configuring Dynamic Access-Internal Routes for DHCP and PPP Subscribers

You can dynamically configure access-internal routes. In releases earlier than Junos OS 15.1, this configuration is optional; if you include it, the values from the access-internal variables are used if the next-hop value is missing in the relevant RADIUS attribute—Framed-Route [22] for IPv4 and Framed-IPv6-Route [99] for IPv6.

Starting in Junos OS Release 15.1R1, we no longer recommend that you always include the `access-internal` stanza in the dynamic-profile when the `access` stanza is present for framed route support. The subscriber's address is stored in the session database entry before the dynamic profile installs the framed route, enabling the next-hop address to be resolved when it is not explicitly specified in the Framed-Route RADIUS attribute (22) or Framed-IPv6-Route attribute [99].

DHCP subscriber interfaces require the `qualified-next-hop` to identify the interface and the MAC address. For PPP subscriber interfaces, you do not need to specify the MAC address for access-internal routes.

To dynamically configure access-internal routes for DHCP or PPP subscribers:

1. Specify that you want to configure the access-internal route.

```
user@host# edit dynamic-profiles profile-name routing-options
```

2. Configure the IP address and the qualified next-hop address as variables.

```
[edit dynamic-profiles profile-name routing-options]
user@host# set access-internal route $junos-subscriber-ip-address qualified-next-hop $junos-
interface-name
```

NOTE: The variable used for qualified-next-hop is \$junos-interface-name.

3. (DHCP subscriber interfaces only) Configure the MAC address for the qualified next-hop as a variable.

```
[edit dynamic-profiles profile-name routing-options access-internal route $junos-subscriber-
ip-address qualified-next-hop $junos-underlying-interface]
user@host# set mac-address $junos-subscriber-mac-address
```

Suppressing DHCP Access, Access-Internal, and Destination Routes

During the DHCP client binding operation, the DHCP process adds route information for the DHCP sessions by default. The DHCP process adds the following routes:

- DHCPv4 sessions—access-internal and destination routes.
- DHCPv6 sessions—access-internal and access routes.

An access route represents a network behind an attached video services router, and is set to a preference of 13.

An access internal route is a /32 route that represents a directly attached end user, and is set to a preference of 12.

These routes are used by the DHCP application on a video services router to represent either the end users or the networks behind the attached video services router.

In some scenarios, you might want to override the default behavior and prevent DHCP from automatically installing the route information.

For example, DHCP relay installs destination (host) routes by default—this action is required in certain configurations to enable address renewals from the DHCP server to work properly. However, the default installation of destination routes might cause a conflict when you configure DHCP relay with static subscriber interfaces.

To avoid such configuration conflicts you can override the default behavior and prevent DHCP relay from installing the routes.

Preventing DHCP from Installing Access, Access-Internal, and Destination Routes by Default

You can use the route suppression option to override the default route installation behavior. You can configure route suppression and prevent DHCP from installing specific types of routes for:

- DHCP local server and DHCP relay agent
- DHCPv4 and DHCPv6 sessions
- Globally or for named interface groups

For DHCPv4 you can override the installation of destination routes only or access-internal routes (the access-internal option prevents installation of both destination and access-internal routes). For DHCPv6 you can specify access routes, access-internal routes, or both.

Example:

- For DHCP local server route suppression (for example, a global configuration):

```
[edit system services dhcp-local-server]
user@host# set route-suppression access-internal
```

- For DHCP relay (for example, a group-specific configuration):

```
[edit forwarding-options dhcp-relay group southeast]
user@host# set route-suppression destination
```

- For DHCPv6 local server (for example, a group-specific configuration):

```
[edit system services dhcp-local-server group southern3]
user@host# set dhcpv6 route-suppression access access-internal
```

- For DHCPv6 relay (for example, a global configuration):

```
[edit forwarding-options dhcp-relay]
user@host# set dhcpv6 route-suppression access
```

Note the following while configuring route suppression option:

- You cannot suppress access-internal routes when the subscriber is configured with both IA_NA and IA_PD addresses over IP demux interfaces—the IA_PD route relies on the IA_NA route for next hop connectivity.
- The no-arp statement supported in legacy DHCP is replaced by the route-suppression statement.

Verifying the Configuration of Access and Access-Internal Routes for DHCP and PPP Subscribers

IN THIS SECTION

- Purpose | 43
- Action | 44

Purpose

View configuration information for access routes and access-internal routes on DHCP and PPP subscribers. The access-internal routes are those that are automatically installed when a client profile is instantiated.

Action

- To display extensive information about access routes and access-internal routes:

```
user@host>show route extensive
```

- To display the configuration for access routes:

```
user@host>show route protocol access
```

- To display the configuration for access-internal routes:

```
user@host> show route protocol access-internal
```

Release History Table

Release	Description
15.1R1	Starting in Junos OS Release 15.1R1, we no longer recommend that you always include the access-internal stanza in the dynamic-profile when the access stanza is present for framed route support.
15.1	Starting in Junos OS Release 15.1, we recommend that you use only access routes for framed route support.
15.1	Starting in Junos OS Release 15.1, we recommend that you use only access routes for framed route support.

RELATED DOCUMENTATION

RADIUS IETF Attributes Supported by the AAA Service Framework

DHCP Overview

DHCPv6 Local Server

DHCPv6 Relay Agent

DHCPv6 Relay Agent Overview

Subscribers with Identical Framed Routes

Subscribers in the same routing instance are typically expected to have different framed routes. However, there is an active/backup use case where you might want to configure the same framed route for two subscribers. In this scenario, a subscriber expects to receive ingress traffic from an active CPE device, but wants to switch as soon as possible to a backup CPE device.

You can meet this requirement by having two subscribers connected to the same BNG with the same access route for an identical Framed-Route address. However, you must configure a distance value in the Framed-Route that is different between the two subscribers. For example, you might configure RADIUS as follows:

```
user1@test.com Cleartext-Password := "$abc123"  
  Framed-Route = "10.0.0.1/32 distance 12",  
  ERX-Virtual-Router-Name = test,  
user2@test.com Cleartext-Password := "$abc123"  
  Framed-Route = "10.0.0.1/32 distance 240",  
  ERX-Virtual-Router-Name = test,
```

Subscribers user1 and user2 have the same password, routing instance, and Framed-Route address. The distance is just an administrative distance or preference for discrimination between the routes. The distance is 12 for user1 and 240 for user2. The router can add only one route to the forwarding table. It selects the route with the lowest distance value, which is 12. Consequently, traffic towards the subscriber travels to the logical interface associated with user1.

The router installs the backup route for user2 in the routing table. If the link to user1 goes down, then the router installs the backup route for user2 in the forwarding table so downstream traffic can continue to the subscriber.

What happens if you do not configure different distance values for the two subscribers? Consider the following RADIUS configuration:

```
user1@test.com Cleartext-Password := "$abc123"  
  Framed-Route = "10.0.0.1/32",  
  ERX-Virtual-Router-Name = test,  
user2@test.com Cleartext-Password := "$abc123"  
  Framed-Route = "10.0.0.1/32",  
  ERX-Virtual-Router-Name = test,
```

If both of these subscribers try to log in, only the one that logs in first achieves the Active state, so only that route is installed in the forwarding table. The other subscriber flaps between the Init and Terminated states and never succeeds at logging in, as long as the first subscriber is Active.

Configuring PWHT on a Transport Logical Interface for BNG

SUMMARY

You can configure PWHT over EVPN-VPWS on a transport logical interface, with subscriber management (BNG) service logical interfaces.

IN THIS SECTION

- [Overview | 46](#)
- [Requirements | 47](#)
- [Configuration | 48](#)

Overview

IN THIS SECTION

- [Topology | 47](#)

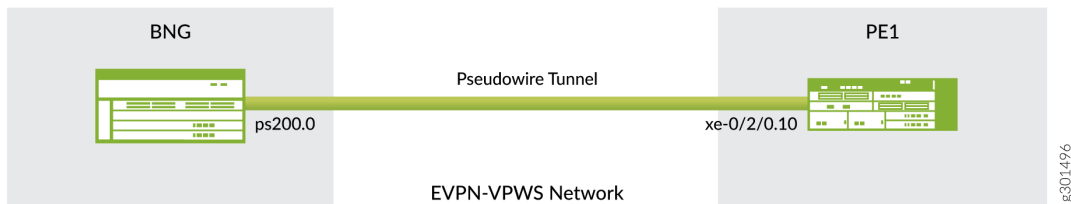
You can deploy a broadband network gateway (BNG) that is connected to an aggregation network running EVPN-VPWS, and you can configure pseudowire headend termination (PWHT) on the transport logical interface that is on the pseudowire subscriber (PS) interface. The BNG pops the EVPN and VPWS headers and terminates subscribers at Layer 2.

This feature includes support for:

- All broadband features available on PWHT on MX Series routers
- Single-homed EVPN-VPWS with the PS interface anchored to a logical tunnel (LT) interface
- Choice of whether or not to use a control word

Topology

This example shows configuration of a single-homed EVPN-VPWS with the PS interface anchored to a logical tunnel (LT) interface.



Requirements

- An MX Series router to serve as the BNG router
- A router (PE1 in the topology) to serve as the EVPN-VPWS remote peer to the BNG router
- Juno OS Release 21.1R1 or later

Before You Begin

This example shows the connection between the BNG router and the EVPN-VPWS remote peer router, PE1. For details on configuring your subscriber management setup—such as CoS dynamic profiles and router advertisement, DHCP or PPPoE clients, RADIUS servers—or your EVPN-VPWS network, see guides such as:

- [Broadband Subscriber Sessions User Guide](#)
- [Broadband Subscriber Services User Guide](#)
- [EVPN User Guide](#)

NOTE: Ensure that you have done these two things before you try to commit the configurations for the two routers:

- Defined the dynamic profile in the [edit dynamic-profiles] hierarchy

- Enabled MPLS to run EVPN instances, so that you can commit the commands that are in the [edit routing-instances VLL_VPWS_PWHT protocols] hierarchy

If you don't have those items defined and enabled when you try to commit the two router configurations, commit errors occur.

Configuration

IN THIS SECTION

- [BNG Router | 48](#)
- [PE1 EVPN-VPWS Remote Peer | 52](#)

Configure the connection between the transport logical interface on the PS interface on the BNG router and the logical interface at the pseudowire tunnel end on the PE1 remote peer router.

BNG Router

Prerequisites

As one of the prerequisites, the `auto-vlan-pwht` must be configured. For example, you can configure PPPoE over auto-configured stacked VLAN. The sample configuration is as follows:

```
[edit routing-instances]
user@host# show auto-vlan-pwht
vlan-demux {
  interfaces {
    demux0 {
      unit "$junos-interface-unit" {
        no-traps;
        vlan-tags outer "$junos-stacked-vlan-id" inner "$junos-vlan-id";
        advisory-options {
          upstream-rate 1g;
          downstream-rate 1g;
        }
      }
    }
  }
}
```

```
    }
    demux-options {
        underlying-interface "$junos-interface-ifd-name";
    }
    family pppoe {
        duplicate-protection;
        dynamic-profile pppv6p;
        max-sessions 1;
        short-cycle-protection {
            lockout-time-min 5;
            lockout-time-max 60;
        }
    }
}
}
```

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set interfaces lt-1/0/0 hierarchical-scheduler maximum-hierarchy-levels 2
set interfaces ps0 description BNG-ps0____PE-xe-0/2/0
set interfaces ps0 anchor-point lt-1/0/0
set interfaces ps0 flexible-vlan-tagging
set interfaces ps0 auto-configure stacked-vlan-ranges dynamic-profile auto-vlan-pwht accept
pppoe ranges any,any
set interfaces ps0 unit 0 encapsulation ethernet-ccc
set routing-instances VLL_VPWS_PWHT instance-type evpn-vpws interface ps0.0
set routing-instances VLL_VPWS_PWHT route-distinguisher 10.255.0.1:100
set routing-instances VLL_VPWS_PWHT vrf-target target:100:1004
set routing-instances VLL_VPWS_PWHT protocols evpn interface ps0.0 vpws-service-id local 33
remote 34
```

Step-by-Step Procedure

1. Navigate to the interfaces hierarchy. Specify the logical tunnel interface that is the anchor point for the pseudowire logical interface device. The anchor point must be an It device in the format *lt-fpc/pic/port*.

```
[edit interfaces]
user@host# set lt-1/0/0 hierarchical-scheduler maximum-hierarchy-levels 2
```

2. Still in the interfaces hierarchy, configure the pseudowire subscriber (PS) interface with the description that you supply and then associate it with its anchor point logical tunnel interface.

```
[edit interfaces]
user@host# set ps0 description BNG-ps0____PE-xe-0/2/0
user@host# set ps0 anchor-point lt-1/0/0
```

3. Configure the VLAN tagging method for mixed (flexible) VLAN tagging on the PS interface.

```
[edit]
user@host# set ps0 flexible-vlan-tagging
```

4. Configure the PS interface to use a dynamic profile when the dynamic VLANs are created. The dynamic profile uses the VLAN ranges configured for the interface.

```
[edit interfaces]
user@host# set ps0 auto-configure stacked-vlan-ranges dynamic-profile auto-vlan-pwht accept
pppoe ranges any,any
```

5. Configure the logical interface for the PS interface (this is the transport logical interface), and configure ethernet-ccc encapsulation.

```
[edit interfaces]
user@host# set ps0 unit 0 encapsulation ethernet-ccc
```

6. Navigate to the `routing-instances` hierarchy, and then configure the routing-instance name, instance type `evpn-vpws`, the route distinguisher, and the VPN routing and forwarding (VRF) target community for the EVPN-VPWS routing instance.

```
[edit routing-instances]
user@host# set VLL_VPWS_PWHT instance-type evpn-vpws interface ps0.0
user@host# set VLL_VPWS_PWHT route-distinguisher 10.255.0.1:1004
user@host# set VLL_VPWS_PWHT vrf-target target:100:1004
```

7. Still in the `routing-instances` hierarchy, configure the interface of the routing instance with local and remote service identifiers. These identifiers identify the PE routers that forward and receive the traffic in the EVPN-VPWS network. The local service identifier is used to identify the PE router that is forwarding the traffic, and the remote service identifier is used to identify the PE router that is receiving the traffic in the network.

```
[edit routing-instances]
user@host# set VLL_VPWS_PWHT protocols evpn interface ps0.0 vpws-service-id local 33 remote 34
```

Results

Check the results of the configuration:

```
[edit interfaces]
user@host# show
lt-2/0/0 {
    hierarchical-scheduler maximum-hierarchy-levels 2;
}
ps0 {
    description BNG-ps0____PE-xe-0/2/0;
    anchor-point {
        lt-2/0/0;
    }
    flexible-vlan-tagging;
    auto-configure {
        stacked-vlan-ranges {
            dynamic-profile auto-vlan-pwht {
                accept pppoe;
                ranges {
                    any,any;
                }
            }
        }
    }
}
```

```

    }
  }
}

unit 0 {
  encapsulation ethernet-ccc;
}
}
[edit routing-instances]
user@host# show
VLL_VPWS_PWHT {
  protocols {
    evpn {
      interface ps0.0 {
        vpws-service-id {
          local 33;
          remote 34;
        }
      }
    }
  }
  instance-type evpn-vpws;
  interface ps0.0;
  route-distinguisher 10.255.0.1:100;
  vrf-target target:100:1004;
}

```

PE1 EVPN-VPWS Remote Peer

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```

set interfaces xe-0/2/0 description PE-xe-0/2/0____BNG-ps0
set interfaces xe-0/2/0 hierarchical-scheduler implicit-hierarchy
set interfaces xe-0/2/0 no-traps
set interfaces xe-0/2/0 flexible-vlan-tagging

```



```

set interfaces xe-0/2/0 encapsulation flexible-ethernet-services
set interfaces xe-0/2/0 unit 10 description VPWS-PE-BNG-PWHT
set interfaces xe-0/2/0 unit 10 encapsulation vlan-ccc
set interfaces xe-0/2/0 unit 10 vlan-id 10
set interfaces xe-0/2/0 unit 10 output-vlan-map swap tag-protocol-id 0x8100 inner-vlan-id 10
set routing-instances VLL_VPWS_PWHT instance-type evpn-vpws interface xe-0/2/0.10
set routing-instances VLL_VPWS_PWHT route-distinguisher 10.255.0.2:1004
set routing-instances VLL_VPWS_PWHT vrf-target target:100:1004
set routing-instances VLL_VPWS_PWHT protocols evpn interface xe-0/2/0.10 vpws-service-id local
34 remote 33

```

Step-by-Step Procedure

1. Navigate to the interfaces hierarchy. Specify the interface that is the connecting interface on the PE1 EVPN-VPWS remote peer.

```

[edit interfaces]
user@host# set xe-0/2/0

```

2. Still in the interfaces hierarchy, configure the peer connection interface with the description that you supply, hierarchical-scheduler implicit-hierarchy, and no-traps.

```

[edit interfaces]
user@host# set xe-0/2/0 description PE-xe-0/2/0____BNG-ps0
user@host# set xe-0/2/0 hierarchical-scheduler implicit-hierarchy
user@host# set xe-0/2/0 no-traps

```

3. Configure the VLAN tagging method for mixed (flexible) VLAN tagging on the connecting interface.

```

[edit interfaces]
user@host# set xe-0/2/0 flexible-vlan-tagging

```

4. Configure flexible-ethernet-services encapsulation on the interface.

```

[edit interfaces]
user@host# set xe-0/2/0 encapsulation flexible-ethernet-services

```

5. Configure the logical interface for the xe-0/2/0 interface , and configure a description, encapsulation, the VLAN ID, and the output-vlan-map settings.

```
[edit interfaces]
user@host# set xe-0/2/0 unit 10 description VPWS-PE-BNG-PWHT
user@host# set xe-0/2/0 unit 10 encapsulation vlan-ccc
user@host# set xe-0/2/0 unit 10 vlan-id 10
user@host# set xe-0/2/0 unit 10 output-vlan-map swap tag-protocol-id 0x8100 inner-vlan-id 10
```

6. Navigate to the routing-instances hierarchy, and then configure the routing-instance name, instance type evpn-vpws, the route distinguisher, and the VPN routing and forwarding (VRF) target community for the EVPN-VPWS routing instance.

```
[edit routing-instances]
user@host# set VLL_VPWS_PWHT instance-type evpn-vpws
user@host# set VLL_VPWS_PWHT interface xe-0/2/0.10
user@host# set VLL_VPWS_PWHT route-distinguisher 10.255.0.2:1004
user@host# set VLL_VPWS_PWHT vrf-target target:100:1004
```

7. Still in the routing-instances hierarchy, configure the interface of the routing instance with local and remote service identifiers. These identifiers identify the PE routers that forward and receive the traffic in the EVPN-VPWS network. The local service identifier is used to identify the PE router that is forwarding the traffic, and the remote service identifier is used to identify the PE router that is receiving the traffic in the network.

```
[edit routing-instances]
user@host# set VLL_VPWS_PWHT protocols evpn interface xe-0/2/0.10 vpws-service-id local 34
remote 33
```

Results

Check the results of the configuration:

```
[edit interfaces]
user@host# show
xe-0/2/0 {
    description PE-xe-0/2/0____BNG-ps0;
```

```
no-traps;
hierarchical-scheduler implicit-hierarchy;
flexible-vlan-tagging;
encapsulation flexible-ethernet-services;
unit 10 {
    description VPWS-PE-BNG-PWHT;
    encapsulation vlan-ccc;
    vlan-id 10;
    output-vlan-map {
        swap;
        tag-protocol-id 0x8100;
        inner-vlan-id 10;
    }
}
}
[edit routing-instances]
user@host# show
VLL_VPWS_PWHT {
    protocols {
        evpn {
            interface xe-0/2/0.10 {
                vpws-service-id {
                    local 34;
                    remote 33;
                }
            }
        }
    }
    instance-type evpn-vpws;
    interface xe-0/2/0.10;
    route-distinguisher 10.255.0.2:1004;
    vrf-target target:100:1004;
}
```

2

CHAPTER

DHCP Subscriber Access Networks

[DHCP Subscriber Access Networks Overview | 57](#)

[DHCP Snooping for Network Security | 65](#)

[DHCPv4 Duplicate Client Management | 92](#)

[DHCPv6 Duplicate Client Management | 98](#)

DHCP Subscriber Access Networks Overview

IN THIS SECTION

- [DHCP and Subscriber Management Overview | 57](#)
- [Subscriber Access Operation Flow Using DHCP Relay | 59](#)
- [Defining Various Levels of Services for DHCP Subscribers | 60](#)
- [Example: Configuring a Tiered Service Profile for Subscriber Access | 61](#)

DHCP and Subscriber Management Overview

IN THIS SECTION

- [Extended DHCP Local Server and Subscriber Management Overview | 58](#)
- [Extended DHCP Relay and Subscriber Management Overview | 58](#)
- [DHCP Relay Proxy and Subscriber Management Overview | 58](#)
- [Extended DHCP over Dynamic PPPoE Subscriber \(IPv4, IPV6, and Dual stack\) Interfaces \(ACX7100 Devices\) | 58](#)

You use DHCP in broadband access networks to provide IP address configuration and service provisioning. DHCP, historically a popular protocol in LANs, works well with Ethernet connectivity and is becoming increasingly popular in broadband networks as a simple, scalable solution for assigning IP addresses to subscriber home PCs, set-top boxes (STBs), and other devices.

Junos OS subscriber management supports the following DHCP allocation models:

- DHCP Local Server
- DHCP Relay
- DHCP Relay Proxy

DHCP uses address assignment pools from which to allocate subscriber addresses. Address-assignment pools support both dynamic and static address assignment:

- Dynamic address assignment—A subscriber is automatically assigned an address from the address-assignment pool.
- Static address assignment—Addresses are reserved and always used by a particular subscriber.

NOTE: Addresses that are reserved for static assignment are removed from the dynamic address pool and cannot be assigned to other clients.

Extended DHCP Local Server and Subscriber Management Overview

You can enable the services router to function as an extended DHCP local server. As an extended DHCP local server the services router, and not an external DHCP server, provides an IP address and other configuration information in response to a client request. The extended DHCP local server supports the use of external AAA authentication services, such as RADIUS, to authenticate DHCP clients.

Extended DHCP Relay and Subscriber Management Overview

You can configure extended DHCP relay options on the router and enable the router to function as a DHCP relay agent. A DHCP relay agent forwards DHCP request and reply packets between a DHCP client and a DHCP server. You can use DHCP relay in carrier edge applications such as video and IPTV to obtain configuration parameters, including an IP address, for your subscribers. The extended DHCP relay agent supports the use of external AAA authentication services, such as RADIUS, to authenticate DHCP clients.

DHCP Relay Proxy and Subscriber Management Overview

DHCP relay proxy mode is an enhancement to extended DHCP relay. DHCP relay proxy supports all DHCP relay features while providing additional features and benefits. Except for the ability to add DHCP relay agent options and the gateway address (giaddr) to DHCP packets, DHCP relay is transparent to DHCP clients and DHCP servers, and simply forwards messages between DHCP clients and servers. When you configure DHCP relay to operate in proxy mode, the relay is no longer transparent. In proxy mode, DHCP relay conceals DHCP server details from DHCP clients, which interact with a DHCP relay in proxy mode as though it is the DHCP server. For DHCP servers there is no change, because proxy mode has no effect on how the DHCP server interacts with the DHCP relay.

Extended DHCP over Dynamic PPPoE Subscriber (IPv4, IPV6, and Dual stack) Interfaces (ACX7100 Devices)

Support is provided for the DDOS protocol group 'dhcpv4v6' on ACX7100 which is a combined DDOS policer for the aggregated traffic of BBE protocols - DHCPv4, DHCPv6, PPPoE, PPP and L2TP.

- DHCP (IP-DEMUX lite) & PPPoE subscribers (IPv4, IPV6, and Dual stack) with CoS Lawful Intercept and filter support.

DVLAN (Single and dual tag) with L2TP (LAC) DDOS policers configuration for BBE protocols. Individual BBE protocol and DDOS protocol group configuration is enabled. Only the aggregate DDOS protocol group 'dhcpv4v6' is active.

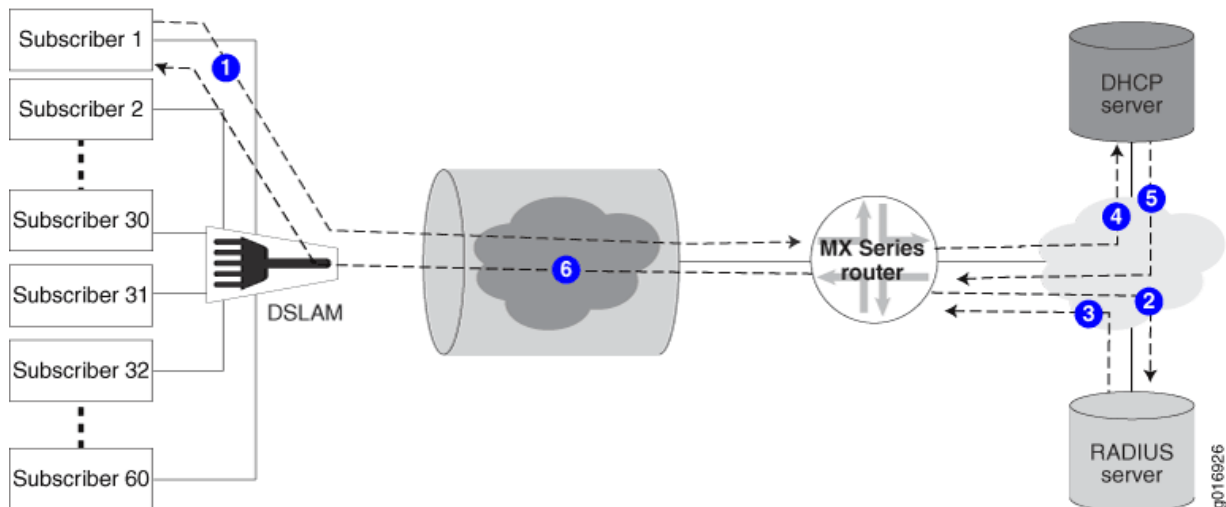
Subscriber scale qualification with Class of Support (CoS) for Layer 2 Tunneling Protocol (L2TP), L2TP access concentrator (LAC) Subscriber Interfaces for IPV4, IPV6 and dual stack.

Subscriber Access Operation Flow Using DHCP Relay

The subscriber management feature requires that a subscriber (for example, a DHCP client) send a discover message to the router interface to initialize dynamic configuration of that interface.

Figure 8 on page 59 shows the flow of operations that occurs when the router is using DHCP relay to enable access for a subscriber.

Figure 8: Subscriber Access Operation Flow



The following general sequence occurs during access configuration for a DHCP client:

1. The client issues a DHCP discover message.
2. The router issues an authorization request to the RADIUS server.
3. The RADIUS server issues an authorization response to the router.
4. The router passes the DHCP discover message through to the DHCP server.

5. The DHCP server issues an IP address for the client.
6. The router DHCP component sends an acknowledgement back to the client.

The subscriber now has access to the network and the authorized service.

Defining Various Levels of Services for DHCP Subscribers

This topic discusses how to create dynamic profiles to define various levels of service for DHCP clients.

Before you configure dynamic profiles for client services:

1. Create a basic dynamic profile.

See *Configuring a Basic Dynamic Profile*.

2. Configure a dynamic profile that enables DHCP clients access to the network.

See *Configuring Dynamic DHCP Client Access to a Multicast Network*

NOTE: You can create a basic dynamic profile that contains both access configuration and some level of basic service.

3. Ensure that the router is configured to enable communication between the client and the RADIUS server.

See *Specifying the Authentication and Accounting Methods for Subscriber Access*.

4. Configure all RADIUS values that you want the profiles to use when validating DHCP clients.

See *RADIUS Servers and Parameters for Subscriber Access*

To configure an initial client access dynamic profile:

1. Access the desired service profile.

```
user@host# set dynamic-profiles basic-service-profile
```

2. (Optional) Define any IGMP protocols values as described for creating a basic access profile to combine a basic service with access in a profile.

See *Configuring Dynamic DHCP Client Access to a Multicast Network*.

3. (Optional) Specify any filters for the interface.

See *Dynamically Attaching Statically Created Filters for Any Interface Type*, *Dynamically Attaching Statically Created Filters for a Specific Interface Family Type*, or *Dynamically Attaching Filters Using RADIUS Variables*.

4. Define any CoS values for the service level you want this profile to configure on the interface.

Example: Configuring a Tiered Service Profile for Subscriber Access

This example shows how to configure a tiered service profile for subscribers.

The profile contains three services:

- Gold—Subscribers that pay for this service are allocated 10M bandwidth for data, voice, and video services.
- Silver—Subscribers that pay for this service are allocated 5M bandwidth for data, voice, and video services.
- Bronze—Subscribers that pay for this service are allocated 1M bandwidth for the data service only.

Each subscriber is allocated a VLAN that is created statically. Subscribers log in using DHCP and authenticate using RADIUS. The subscribers can migrate from one service to another when they change subscriptions.

To configure a profile for a tiered service:

1. Configure the VLAN interfaces associated with each subscriber. Enable hierarchical scheduling for the interface.

```

interfaces {
  ge-2/0/0 {
    description subscribers;
    hierarchical-scheduler;
    stacked-vlan-tagging;
    unit 1 {
      vlan-tags outer 100 inner 100;
      family inet {
        unnumbered-address lo0.0 preferred-source-address 127.0.0.2;
      }
    }
    unit 2 {
      family inet {
        vlan-tags outer 101 inner 101;
      }
    }
  }
}

```

```

        unnumbered-address lo0.0 preferred-source-address 127.0.0.2;
    }
}
unit 3 {
    vlan-tags outer 102 inner 102;
    family inet {
        unnumbered-address lo0.0 preferred-source-address 127.0.0.2;
    }
}
}
}

```

2. Configure the static CoS parameters.

In this example, each offering (video, voice, and data) is assigned a queue, and each service (Gold, Silver, and Bronze) is assigned a scheduler.

```

class-of-service {
    forwarding-classes {
        queue 0 data;
        queue 1 voice;
        queue 2 video;
    }
    scheduler-maps {
        bronze_service_smap {
            forwarding-class data scheduler data_sch;
        }
        silver_service_smap {
            forwarding-class data scheduler data_sch;
            forwarding-class voice scheduler silver_voice_sch;
            forwarding-class video scheduler silver_video_sch;
        }
        gold_service_smap {
            forwarding-class data scheduler data_sch;
            forwarding-class voice scheduler gold_voice_sch;
            forwarding-class video scheduler gold_video_sch;
        }
    }
}
schedulers {
    data_sch {
        transmit-rate percent 20;
        buffer-size remainder;
    }
}

```

```

        priority low;
    }
    silver_voice_sch {
        transmit-rate percent 30;
        buffer-size remainder;
        priority high;
    }
    silver_video_sch {
        transmit-rate percent 30;
        buffer-size remainder;
        priority medium;
    }
    gold_voice_sch {
        transmit-rate percent 40;
        buffer-size remainder;
        priority high;
    }
    gold_video_sch {
        transmit-rate percent 40;
        buffer-size remainder;
        priority medium;
    }
}
}
}

```

3. Configure the dynamic profile for the service.

The scheduler maps configured for each service are referenced in the dynamic profile.

```

dynamic-profiles {
    subscriber_profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
    }
    class-of-service {
        traffic-control-profiles {
            subscriber_tcp {
                scheduler-map $smap;
            }
        }
    }
}

```

```

        shaping-rate $shaping-rate;
        guaranteed-rate $guaranteed-rate;
        delay-buffer-rate $delay-buffer-rate;
    }
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-underlying-interface-unit" {
            output-traffic-control-profile subscriber_tcp;
        }
    }
}
}

```

4. Configure access for the subscribers.

The DHCP relay agent forwards DHCP request and reply packets between a DHCP client and a DHCP server. You use DHCP relay to obtain configuration parameters, including an IP address, for subscribers. In this example, one DHCP server, address 198.51.100.1, can be used by subscribers.

The DHCP relay configuration is attached to an active server group named `service_provider_group`.

The subscribers are grouped together within the `subscriber_group`, and identifies characteristics such as authentication, username info, and the associated interfaces for the group members. In this example, it also identifies the active server group and the dynamic interface that is used by the subscribers in the group.

```

forwarding-options {
    dhcp-relay {
        server-group {
            service_provider_group {
                198.51.100.1;
            }
        }
    }
    group subscriber_group {
        active-server-group service_provider_group;
        dynamic-profile subscriber_profile;
        interface ge-2/0/0.1;
        interface ge-2/0/0.2;
        interface ge-2/0/0.3;
    }
}

```

```
}  
}
```

RELATED DOCUMENTATION

DHCP Overview

DHCPv6 Local Server

DHCPv6 Relay Agent

Address-Assignment Pools for Subscriber Management

Introduction to Subscriber Management

Dynamic Profiles for Subscriber Management

CoS for Subscriber Access Overview

DHCP Snooping for Network Security

IN THIS SECTION

- [DHCP Snooping Support | 66](#)
- [Configuring DHCP Snooped Packets Forwarding Support for DHCP Local Server | 69](#)
- [Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent | 70](#)
- [Configuring DHCP Snooped Packets Forwarding Support for DHCP Relay Agent | 77](#)
- [Disabling DHCP Snooping Filters | 80](#)
- [Example: Configuring DHCP Snooping Support for DHCP Relay Agent | 82](#)
- [Example: Enabling DHCP Snooping Support for DHCPv6 Relay Agent | 85](#)
- [Preventing DHCP Spoofing | 90](#)

DHCP Snooping Support

IN THIS SECTION

- [What is DHCP Snooping | 66](#)
- [Benefits of DHCP Snooping | 66](#)
- [Activating DHCP Snooping | 66](#)
- [Configuring DHCP Snooping | 67](#)

DHCP snooping provides additional security by identifying the incoming DHCP packets and rejecting DHCP traffic determined to be unacceptable from untrusted devices in the network.

What is DHCP Snooping

DHCP allocates IP addresses dynamically, leasing addresses to devices so that the addresses can be reused when they are no longer needed by the devices to which they were assigned. Hosts and end devices that require IP addresses obtained through DHCP must communicate with a DHCP server across the LAN.

DHCP snooping looks into incoming DHCP packets and examines DHCP messages. It extracts their IP addresses and lease information allocated to clients and builds up a database. Using this database, it can determine if the packets arriving are from the valid clients—that is—the IP addresses of the clients was assigned by the DHCP server. In this way, DHCP snooping acts as a guardian of network security by keeping track of valid IP addresses assigned to downstream network devices by a trusted DHCP server (the server is connected to a trusted network port).

Benefits of DHCP Snooping

- DHCP snooping provides an extra layer of security via dynamic IP source filtering.
- DHCP snooping can prevent rogue DHCP activity in the network by filtering out DHCP packets that are arriving on the wrong ports, or with incorrect contents.

Activating DHCP Snooping

When you are using the DHCP snooping feature, it is important that you understand about enabling the DHCP snooping feature.

On Junos OS device, you cannot configure DHCP snooping feature as an independent feature. Whenever you configure DHCP security or DHCP relay or DHCP server for a specific VLAN or interface or routing instance of the device, the DHCP snooping is automatically enabled on that VLAN/interface/routing instance to perform its task.

For example:

- When you enable DHCP Relay on a given list of interfaces of a specific routing instance
- DHCP snooping gets automatically enabled on those interfaces for that routing instance.
- When you enable DHCP security on a specific VLAN, DHCP snooping gets automatically enabled on that VLAN.

Junos OS enables DHCP snooping on a switch/router/firewall in:

- A routing instance when you configure the following options in that routing instance:
 - dhcp-relay statement at the [edit forwarding-options] hierarchy level.
 - dhcp-local-server statement at the [edit system services] hierarchy level.
- A switch when you configure the following option for any port security features:
 - dhcp-security statement at the [edit vlans vlan-name forwarding-options] hierarchy level.

TIP: If you need to configure DHCP relay, use the forward-only statement unless you need subscriber management or class-of-service (CoS).

We recommend you read the DHCP documentation and use a lab with DHCP traceoptions enabled to check and understand the configuration.

Configuring DHCP Snooping

In the default DHCP snooping configuration, all traffic is snooped.

On Junos OS device, DHCP snooping is enabled in a routing instance when you configure the following options in that routing instance:

- dhcp-relay statement at the [edit forwarding-options] hierarchy level
- dhcp-local-server statement at the [edit system services] hierarchy level
- You can optionally use the forward-snooped-clients statement to evaluate the snooped traffic and to determine if the traffic is forwarded or dropped, based on whether or not the interface is configured as part of a group.

The router discards snooped packets by default if there is no subscriber associated with the packet. To enable normal processing of snooped packets, you must explicitly configure the `allow-snooped-clients` statement at the `[edit forwarding-options dhcp-relay]` hierarchy level.

You can configure DHCP snooping support for a specific routing instance for the following:

- DHCPv4 relay agent—Override the router's (or switch's) default snooping configuration and specify that DHCP snooping is enabled or disabled globally, for a named group of interfaces, or for a specific interface within a named group.

In a separate procedure, you can set a global configuration to specify whether the DHCPv4 relay agent forwards or drops snooped packets for all interfaces, only configured interfaces, or only nonconfigured interfaces. The router also uses the global DHCP relay agent snooping configuration to determine whether to forward or drop snooped BOOTREPLY packets. A renew request may be unicast directly to the DHCP server. This is a BOOTPREQUEST packet and is snooped.

- DHCPv6 relay agent—As you can with snooping support for the DHCPv4 relay agent, you can override the default DHCPv6 relay agent snooping configuration on the router to explicitly enable or disable snooping support globally, for a named group of interfaces, or for a specific interface with a named group of interfaces.

In multi-relay topologies where more than one DHCPv6 relay agent is between the DHCPv6 client and the DHCPv6 server, snooping enables intervening DHCPv6 relay agents between the client and the server to correctly receive and process the unicast traffic from the client and forward it to the server. The DHCPv6 relay agent snoops incoming unicast DHCPv6 packets by setting up a filter with UDP port 547 (the DHCPv6 UDP server port) on a per-forwarding table basis. The DHCPv6 relay agent then processes the packets intercepted by the filter and forwards the packets to the DHCPv6 server.

Unlike the DHCPv4 relay agent, the DHCPv6 relay agent does not support global configuration of forwarding support for DHCPv6 snooped packets.

- DHCP local server—Configure whether DHCP local server forwards or drops snooped packets for all interfaces, only configured interfaces, or only nonconfigured interfaces.
- You can also disable snooping filters. In the preceding configurations, all DHCP traffic is forwarded to the slower routing plane of the routing instance before it is either forwarded or dropped. Disabling snooping filters causes DHCP traffic that can be forwarded directly from the faster hardware control plane to bypass the routing control plane.

Configuring DHCP Snooped Packets Forwarding Support for DHCP Local Server

You can configure how DHCP local server handles DHCP snooped packets. Depending on the configuration, DHCP local server either forwards or drops the snooped packets it receives.

Table 2 on page 69 indicates the action the router takes for DHCP local server snooped packets.

NOTE: Configured interfaces are those interfaces that have been configured with the **group** statement in the [edit system services dhcp-local-server] hierarchy. Non-configured interfaces are those that are in the logical system/routing instance but have not been configured by the **group** statement.

Table 2: Actions for DHCP Local Server Snooped Packets

forward-snooped-clients Configuration	Action on Configured Interfaces	Action on Non-Configured Interfaces
forward-snooped-clients not configured	dropped	dropped
all-interfaces	forwarded	forwarded
configured-interfaces	forwarded	dropped
non-configured-interfaces	dropped	forwarded

To configure DHCP snooped packet forwarding for DHCP local server:

1. Specify that you want to configure DHCP local server.

```
[edit]
user@host# edit system services dhcp-local-server
```

2. Enable DHCP snooped packet forwarding for DHCP local server.

```
[edit system services dhcp-local-server]
user@host# edit forward-snooped-clients
```

3. Specify the interfaces that are supported for snooped packet forwarding.

```
[edit system services dhcp-local-server forward-snooped-clients]
user@host# set (all-interfaces | configured-interfaces | non-configured-interfaces)
```

For example, to configure DHCP local server to forward DHCP snooped packets on only configured interfaces:

```
[edit]
system {
  services {
    dhcp-local-server {
      forward-snooped-clients configured-interfaces;
    }
  }
}
```

SEE ALSO

| [DHCP Snooping Support](#)

Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent

DHCP relay agent uses a two-part configuration to determine how to handle DHCP snooped packets. This topic describes the first procedure, in which you enable or disable snooping support for DHCP relay agent and, optionally, override the default snooping configuration.

The second procedure, which applies only to DHCPv4 relay agent, is described in [Configuring DHCP Snooped Packets Forwarding Support for DHCP Relay Agent](#), and configures the forwarding action for snooped clients, which specifies whether DHCP relay agent forwards or drops snooped traffic.

You can enable or disable DHCP globally for DHCP relay, for a group of interfaces, or for a specific interface in a group.

By default, DHCP snooping is enabled for DHCP relay. To enable or disable DHCP snooping support globally:

1. Specify that you want to configure DHCP relay agent.

- For DHCP relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

- For DHCPv6 relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay dhcpv6
```

2. Specify that you want to override the default configuration.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay]
user@host# edit overrides
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# edit overrides
```

3. Enable or disable DHCP snooping support.

- To enable DHCP snooping:

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay overrides]
user@host# set allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 overrides]
user@host# set allow-snooped-clients
```

- To disable DHCP snooping:
 - For DHCP relay agent:

```
[edit forwarding-options dhcp-relay overrides]
user@host# set no-allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 overrides]
user@host# set no-allow-snooped-clients
```

For example, to enable global DHCP snooping support :

```
forwarding-options {
  dhcp-relay {
    overrides {
      allow-snooped-clients;
    }
  }
}
```

To enable or disable DHCP snooping support for a group of interfaces:

1. Specify that you want to configure DHCP relay agent.

- For DHCP relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

- For DHCPv6 relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay dhcpv6
```

2. Specify the named group.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay]
user@host# edit group group-name
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# edit group group-name
```

3. Specify that you want to override the default configuration.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name]
user@host# edit overrides
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name]
user@host# edit overrides
```

4. Enable or disable DHCP snooping support.

- To enable DHCP snooping:
 - For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name overrides]
user@host# set allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name overrides]
user@host# set allow-snooped-clients
```

- To disable DHCP snooping:

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name overrides]
user@host# set no-allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name overrides]
user@host# set no-allow-snooped-clients
```

For example, to enable DHCP snooping support on all interfaces in group boston:

```
forwarding-options {
  dhcp-relay {
    group boston {
      overrides {
        allow-snooped-clients;
      }
    }
  }
}
```

To enable or disable DHCP snooping support on a specific interface:

1. Specify that you want to configure DHCP relay agent.

- For DHCP relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

- For DHCPv6 relay agent:

```
[edit]
user@host# edit forwarding-options dhcp-relay dhcpv6
```

2. Specify the named group containing the interface.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay]
user@host# edit group group-name
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# edit group group-name
```

3. Specify the interface for which you want to configure DHCP snooping.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name]
user@host# edit interface interface-name
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name]
user@host# edit interface interface-name
```

4. Specify that you want to override the default configuration on the interface.

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name interface interface-name]
user@host# edit overrides
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name interface interface-name]
user@host# edit overrides
```

5. Enable or disable DHCP snooping support.

- To enable DHCP snooping:

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name interface interface-name
overrides]
user@host# set allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name interface interface-name
overrides]
user@host# set allow-snooped-clients
```

- To disable DHCP snooping:

- For DHCP relay agent:

```
[edit forwarding-options dhcp-relay group group-name interface interface-name
overrides]
user@host# set no-allow-snooped-clients
```

- For DHCPv6 relay agent:

```
[edit forwarding-options dhcp-relay dhcpv6 group group-name interface interface-name
overrides]
user@host# set no-allow-snooped-clients
```

For example, to disable DHCP snooping support on interface ge-2/1/8.0 in group boston:

```
forwarding-options {
  dhcp-relay {
    group boston {
      interface ge-2/1/8.0 {
        overrides {
          no-allow-snooped-clients;
        }
      }
    }
  }
}
```



```

    }
}

```

To enable DHCPv6 snooping support on interface ge-3/2/1.1 in group sunnyvale:

```

forwarding-options {
  dhcp-relay {
    dhcpv6 {
      group sunnyvale {
        interface ge-3/2/1.1 {
          overrides {
            allow-snooped-clients;
          }
        }
      }
    }
  }
}

```

SEE ALSO

[DHCP Snooping Support](#)

[Configuring DHCP Snooped Packets Forwarding Support for DHCP Relay Agent](#)

[Example: Configuring DHCP Snooping Support for DHCP Relay Agent | 82](#)

Overriding the Default DHCP Relay Configuration Settings

Configuring DHCP Snooped Packets Forwarding Support for DHCP Relay Agent

You can configure how DHCP relay agent handles DHCP snooped packets. Depending on the configuration, DHCP relay agent either forwards or drops the snooped packets it receives.

DHCP relay uses a two-part configuration to determine how to handle DHCP snooped packets. This topic describes how you use the `forward-snooped-clients` statement to manage whether DHCP relay agent forwards or drops snooped packets, depending on the type of interface on which the packets are snooped. In the other part of the DHCP relay agent snooping configuration, you enable or disable the DHCP relay snooping feature.

Table 3 on page 78 shows the action the router or switch takes on snooped packets when DHCP snooping is enabled by the `allow-snooped-clients` statement.

The router or switch also uses the configuration of the DHCP relay agent forwarding support to determine how to handle snooped BOOTREPLY packets.

Table 3: Actions for DHCP Relay Agent Snooped Packets When DHCP Snooping Is Enabled

<code>forward-snooped-clients</code> Configuration	Action on Configured Interfaces	Action on Non-Configured Interfaces
<code>forward-snooped-clients</code> not configured	snooped packets result in subscriber (DHCP client) creation	dropped
<code>all-interfaces</code>	forwarded	forwarded
<code>configured-interfaces</code>	forwarded	dropped
<code>non-configured-interfaces</code>	snooped packets result in subscriber (DHCP client) creation	forwarded

Table 4 on page 78 shows the action the router (or switch) takes on snooped packets when DHCP snooping is disabled by the `no-allow-snooped-clients` statement.

Table 4: Actions for DHCP Relay Agent Snooped Packets When DHCP Snooping Is Disabled

<code>forward-snooped-clients</code> Configuration	Action on Configured Interfaces	Action on Non-Configured Interfaces
<code>forward-snooped-clients</code> not configured	dropped	dropped
<code>all-interfaces</code>	dropped	forwarded
<code>configured-interfaces</code>	dropped	dropped
<code>non-configured-interfaces</code>	dropped	forwarded

Table 5 on page 79 shows the action the router (or switch) takes for the snooped BOOTREPLY packets.

Table 5: Actions for Snooped BOOTREPLY Packets

forward-snooped-clients Configuration	Action
forward-snooped-clients not configured	snooped BOOTREPLY packets dropped if client is not found
forward-snooped-clients all configurations	snooped BOOTREPLY packets forwarded if client is not found

Configured interfaces have been configured with the **group** statement in the [edit forwarding-options dhcp-relay] hierarchy. Non-configured interfaces are in the logical system/routing instance but have not been configured by the **group** statement.

To configure DHCP snooped packet forwarding and BOOTREPLY snooped packet forwarding for DHCP relay agent:

1. Specify that you want to configure DHCP relay agent.

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

2. Enable DHCP snooped packet forwarding.

```
[edit forwarding-options dhcp-relay]
user@host# edit forward-snooped-clients
```

3. Specify the interfaces that are supported for snooped packet forwarding.

```
[edit forwarding-options dhcp-relay forward-snooped-clients]
user@host# set (all-interfaces | configured-interfaces | non-configured-interfaces)
```

For example, to configure DHCP relay agent to forward DHCP snooped packets on only configured interfaces:

```
[edit]
forwarding-options {
  dhcp-relay {
    forward-snooped-clients configured-interfaces;
```

```
}
}
```

Disabling DHCP Snooping Filters

DHCP snooping provides DHCP security by identifying incoming DHCP packets. In the default DHCP snooping configuration, all traffic is snooped. You can optionally use the `forward-snooped-clients` statement to evaluate the snooped traffic and to determine whether the traffic is forwarded or dropped, based on whether or not the interface is configured as part of a group.

In both the default configuration and in configurations using the `forward-snooped-clients` statement, all DHCP traffic is forwarded from the hardware control plane to the routing plane of the routing instance to ensure that all DHCP packets are intercepted. In certain topologies, such as a Metropolitan Routing Ring topology, forwarding all DHCP traffic to the control plane can result in excessive traffic. The `no-snoop` configuration statement disables the snooping filter for DHCP traffic that can be directly forwarded on the hardware control plane, such as Layer 3 unicast packets with a valid route, causing those DHCP packets to bypass the slower routing plane. You can disable DHCP snooping filters starting in Junos OS Release 15.1R2.

To disable DHCP snooping filters on the DHCP local server:

1. Specify that you want to configure DHCP local server.

```
[edit]
user@host# edit system services dhcp-local-server
```

2. Disable DHCP snooping filters for DHCP local server.

```
[edit system services dhcp-local-server]
user@host# set no-snoop
```

3. Specify that you want to configure DHCPv6 local server.

```
[edit system services dhcp-local-server]
user@host# edit dhcpv6
```

4. Disable DHCP snooping filters for DHCPv6 local server.

```
[edit system services dhcp-local-server dhcpv6]
user@host# set no-snoop
```

To disable DHCP snooping filters on the DHCP relay server:

1. Specify that you want to configure DHCP relay server.

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

2. Disable DHCP snooping filters for DHCP local server.

```
[edit forwarding-options dhcp-relay]
user@host# set no-snoop
```

3. Specify that you want to configure DHCPv6 relay server.

```
[edit forwarding-options dhcp-relay]
user@host# edit dhcpv6
```

4. Disable DHCP snooping filters for DHCPv6 local server.

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# set no-snoop
```

SEE ALSO

[DHCP Snooping Support](#)

no-snoop

Example: Configuring DHCP Snooping Support for DHCP Relay Agent

IN THIS SECTION

- [Requirements | 82](#)
- [Overview | 82](#)
- [Configuration | 82](#)

This example shows how to configure DHCP snooping support for DHCP relay agent.

Requirements

- Configure DHCP relay agent. See *Extended DHCP Relay Agent Overview*.

Overview

In this example, you configure DHCP snooping support for DHCP relay agent by completing the following operations:

- Override the default DHCP snooping configuration and enable DHCP snooping support for the interfaces in group **frankfurt**.
- Configure DHCP relay agent to forward snooped packets to only configured interfaces.

Configuration

IN THIS SECTION

- [Procedure | 82](#)

Procedure

Step-by-Step Procedure

To configure DHCP relay support for DHCP snooping:

1. Specify that you want to configure DHCP relay agent.

```
[edit]
user@host# edit forwarding-options dhcp-relay
```

2. Specify the named group of interfaces on which DHCP snooping is supported.

```
[edit forwarding-options dhcp-relay]
user@host# edit group frankfurt
```

3. Specify the interfaces that you want to include in the group. DHCP relay agent considers these as the configured interfaces when determining whether to forward or drop traffic.

```
[edit forwarding-options dhcp-relay group frankfurt]
user@host# set interface fe-1/0/1.3 upto fe-1/0/1.9
```

4. Specify that you want to override the default configuration for the group.

```
[edit forwarding-options dhcp-relay group frankfurt]
user@host# edit overrides
```

5. Enable DHCP snooping support for the group.

```
[edit forwarding-options dhcp-relay group frankfurt overrides]
user@host# set allow-snooped-clients
```

6. Return to the [edit forwarding-options dhcp-relay] hierarchy level to configure the forwarding action and specify that DHCP relay agent forward snooped packets on only configured interfaces:

```
[edit forwarding-options dhcp-relay group frankfurt overrides]
user@host# up 2
```

7. Enable DHCP snooped packet forwarding for DHCP relay agent.

```
[edit forwarding-options dhcp-relay]
user@host# edit forward-snooped-clients
```

8. Specify that snooped packets are forwarded on only configured interfaces (the interfaces in group frankfurt).

```
[edit forwarding-options dhcp-relay forward-snooped-clients]
user@host# set configured-interfaces
```

Results

From configuration mode, confirm your configuration by entering the `show forwarding-options` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it. The following output also shows a range of configured interfaces in group `frankfurt`.

```
[edit]
user@host# show forwarding-options
dhcp-relay {
  forward-snooped-clients configured-interfaces;
  group frankfurt {
    overrides {
      allow-snooped-clients;
    }
    interface fe-1/0/1.3 {
      upto fe-1/0/1.9;
    }
  }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Example: Enabling DHCP Snooping Support for DHCPv6 Relay Agent

IN THIS SECTION

- Requirements | 85
- Overview | 85
- Configuration | 86
- Verification | 89

Snooping support for DHCPv6 relay agent is disabled on the router by default. This example shows how to override the default DHCPv6 relay agent snooping configuration to explicitly enable DHCPv6 snooping for a named group of interfaces and for a specific interface within a different named group.

NOTE: You can also enable DHCPv6 snooping support globally by using the `allow-snooped-clients` statement at the `[edit forwarding-options dhcp-relay dhcpv6 overrides]` hierarchy level.

Requirements

This example uses the following hardware and software components:

- MX Series 5G Universal Routing Platforms
- Junos OS Release 12.1 or later

Before you begin:

- Configure DHCPv6 relay agent.

See *DHCPv6 Relay Agent Overview*

- Configure named DHCPv6 relay agent interface groups to which you want to apply a common DHCP configuration.

See *Grouping Interfaces with Common DHCP Configurations*

Overview

In this example, you override the default DHCPv6 relay agent snooping configuration to explicitly enable DHCP snooping for both of the following:

- All of the interfaces in the group named boston
- Interface ge-3/2/1.1 in the group named sunnyvale

Configuration

IN THIS SECTION

- [CLI Quick Configuration | 86](#)
- [Enabling DHCPv6 Snooping Support for a Named Group of Interfaces | 86](#)
- [Enabling DHCPv6 Snooping Support for a Specific Interface in a Named Group | 87](#)

To override the default DHCPv6 relay agent snooping configuration to explicitly enable DHCPv6 snooping for a named group of interfaces and for a specific interface within a named group, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set forwarding-options dhcp-relay dhcpv6 group boston overrides allow-snooped-clients
set forwarding-options dhcp-relay dhcpv6 group sunnyvale interface ge-3/2/1.1 overrides allow-snooped-clients
```

Enabling DHCPv6 Snooping Support for a Named Group of Interfaces

Step-by-Step Procedure

To enable DHCPv6 snooping support for a named group of interfaces:

1. Specify that you want to configure DHCPv6 relay agent.

```
[edit]
user@host# edit forwarding-options dhcp-relay dhcpv6
```

2. Specify the named group of interfaces for which you want to enable DHCPv6 snooping.

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# edit group boston
```

3. Specify that you want to override the default DHCPv6 configuration for the interfaces in that group.

```
[edit forwarding-options dhcp-relay dhcpv6 group boston]
user@host# edit overrides
```

4. Enable DHCPv6 snooping support for all interfaces in group boston.

```
[edit forwarding-options dhcp-relay dhcpv6 group boston overrides]
user@host# set allow-snooped-clients
```

Results

From configuration mode, confirm the results of your configuration by issuing the `show` statement at the `[edit forwarding-options dhcp-relay]` hierarchy level. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit forwarding-options dhcp-relay]
user@host# show
dhcpv6 {
  group boston {
    overrides {
      allow-snooped-clients;
    }
  }
}
```

If you are done configuring the router, enter `commit` from configuration mode.

Enabling DHCPv6 Snooping Support for a Specific Interface in a Named Group

Step-by-Step Procedure

To enable DHCPv6 snooping support for a specific interface within a named group of interfaces:

1. Return to the [edit forwarding-options dhcp-relay dhcpv6] hierarchy level to specify that you want to configure DHCPv6 relay agent.

```
[edit forwarding-options dhcp-relay dhcpv6 group boston overrides]
user@host# up 2
```

2. Specify the named group containing the interface.

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# edit group sunnyvale
```

3. Specify the interface in group sunnyvale for which you want to enable DHCPv6 snooping.

```
[edit forwarding-options dhcp-relay dhcpv6 group sunnyvale]
user@host# edit interface ge-3/2/1.1
```

4. Specify that you want to override the default DHCPv6 configuration for interface ge-3/2/1.1 in group sunnyvale.

```
[edit forwarding-options dhcp-relay dhcpv6 group sunnyvale interface ge-3/2/1.1]
user@host# edit overrides
```

5. Enable DHCPv6 snooping support for interface ge-3/2/1.1 in group sunnyvale.

```
[edit forwarding-options dhcp-relay dhcpv6 group sunnyvale interface ge-3/2/1.1 overrides]
user@host# set allow-snooped-clients
```

Results

From configuration mode, confirm the results of your configuration by issuing the show statement at the [edit forwarding-options dhcp-relay] hierarchy level. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit forwarding-options dhcp-relay]
user@host# show
dhcpv6 {
```

```
group boston {
    overrides {
        allow-snooped-clients;
    }
}
group sunnyvale {
    interface ge-3/2/1.1 {
        overrides {
            allow-snooped-clients;
        }
    }
}
}
```

If you are done configuring the router, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- [Verifying the Address Bindings for DHCPv6 Relay Agent Clients | 89](#)

To verify the DHCPv6 configuration in a multi-relay topology, perform this task:

Verifying the Address Bindings for DHCPv6 Relay Agent Clients

Purpose

Verify the DHCPv6 address bindings in the Dynamic Host Configuration Protocol (DHCP) client table.

Action

Display detailed information about address bindings for DHCPv6 relay agent clients.

```
user@host > show dhcpv6 relay binding detail
```

```
Session Id: 13
```

```
Client IPv6 Prefix:                2001:db8:0:8001::5/128
```

```
Client DUID:                        LL0x1-00:00:5e:00:53:02
```

```

State:                                BOUND(DHCPV6_RELAY_STATE_BOUND)
Lease Expires:                        2011-11-21 06:14:50 PST
Lease Expires in:                     293 seconds
Lease Start:                          2011-11-21 06:09:50 PST
Incoming Client Interface:            ge-3/2/1.1
Server Address:                       unknown
Next Hop Server Facing Relay:         2001:db8::2
Server Interface:                     none
Client Id Length:                     10
Client Id:                            /0x00030001/0x00006503/0x0102

```

Meaning

The `Server Address` field in the `show dhcpv6 relay binding detail` command output typically displays the IP address of the DHCPv6 server. In this example, the value `unknown` in the `Server Address` field indicates that this is a multi-relay topology in which the DHCPv6 relay agent is not directly adjacent to the DHCPv6 server, and does not detect the IP address of the server.

In that case, the output instead includes the `Next Hop Server Facing Relay` field, which displays the next-hop address in the direction of the DHCPv6 server.

SEE ALSO

[DHCPv6 Relay Agent Overview](#)

[DHCP Snooping Support](#)

[Grouping Interfaces with Common DHCP Configurations](#)

[Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent | 70](#)

Preventing DHCP Spoofing

A problem that sometimes occurs with DHCP is *DHCP spoofing*. In DHCP spoofing, an untrusted client floods a network with DHCP messages. Often these attacks utilize source IP address spoofing to conceal the true source of the attack.

DHCP snooping helps prevent DHCP spoofing by copying DHCP messages to the control plane and using the information in the packets to create anti-spoofing filters. The anti-spoofing filters bind a client's MAC address to its DHCP-assigned IP address and use this information to filter spoofed DHCP messages. In a typical topology, a carrier edge router (in this function also referred to as the broadband

network gateway [BNG]) connects the DHCP server and the MX Series router (or broadband services aggregator [BSA]) performing the snooping. The MX Series router connects to the client and the BNG.

To configure DHCP snooping, you include the appropriate interfaces within a DHCP group. You can configure DHCP snooping for VPLS environments and bridge domains.

- In a VPLS environment, DHCP requests are forwarded over pseudowires. You configure DHCP snooping over VPLS at the `[edit routing-instances routing-instance-name]` hierarchy level.
- In bridge domains, DHCP snooping works on a per learning bridge basis. Each learning domain must have an upstream interface configured. This interface acts as the flood port for DHCP requests coming from the client side. DHCP requests are forwarded across learning domains in a bridge domain. You configure DHCP snooping on bridge domains at the `[edit routing-instances routing-instance-name bridge-domains bridge-domain-name]` hierarchy level.

To configure DHCP relay to prevent DHCP spoofing:

1. Access the appropriate hierarchy for either a VPLS or bridge domain configuration.

```
user@host# edit routing-instances blue
```

2. Specify that you want to configure DHCP relay.

```
[edit routing-instances blue]
user@host# edit forwarding-options dhcp-relay
```

3. Create the group and assign a name.

```
[edit routing-instances blue forwarding-options dhcp-relay]
user@host# edit group svl-10
```

4. Specify the names of one or more interfaces. DHCP will trust only the MAC addresses learned on the specified interfaces.

```
[edit routing-instances blue forwarding-options dhcp-relay group svl-10]
user@host# set interface fe-1/0/1.1
user@host# set interface fe-1/0/1.2
```

NOTE: You can explicitly enable and disable interface support for DHCP snooped clients. See ["Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent"](#) on page 70.

SEE ALSO

Extended DHCP Relay Agent Overview

[Enabling and Disabling DHCP Snooped Packets Support for DHCP Relay Agent | 70](#)

Release History Table

Release	Description
15.1R2	You can disable DHCP snooping filters starting in Junos OS Release 15.1R2.

DHCPv4 Duplicate Client Management

IN THIS SECTION

- [DHCPv4 Duplicate Client In Subnet Overview | 92](#)
- [Guidelines for Configuring Support for DHCPv4 Duplicate Clients | 93](#)
- [Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Option 82 Information | 94](#)
- [Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Their Incoming Interfaces | 96](#)

DHCPv4 Duplicate Client In Subnet Overview

In some network environments, client IDs and hardware addresses (MAC addresses) might not be unique, resulting in duplicate clients. A duplicate DHCP client occurs when a client attempts to get a lease, and that client has the same client ID or the same hardware address as an existing DHCP client—

the existing client and the new client cannot exist simultaneously, unless you have configured the optional duplicate client support.

By default, DHCP local server and DHCP relay agent use the subnet information to differentiate between duplicate clients. However, in some cases, this level of differentiation is not adequate. For example, when multiple subinterfaces share the same underlying loopback interface with the same preferred source address, the interfaces appear to be on the same subnet.

You can enable support for duplicate clients in a subnet by configuring DHCP to use additional information to uniquely identify clients—the additional information is either the client incoming interface or the option 82 information in the DHCP packets. Using the option 82 information provides the following important benefits:

- You can configure DHCP relay to preserve and use the remotely created option 82.
- DHCP local server can support an environment in which an aggregation device is present between the client and the DHCP server.

When configured to support duplicate clients in the subnet, DHCP uses the following information to distinguish between the duplicate clients:

- The subnet on which the client resides
- The client ID or hardware address
- The duplicate clients option you configure—either the client incoming interface or the option 82 information in the client's incoming DHCP packets

NOTE: Starting in Junos OS Release 16.1R5, 16.2R2, 17.1R2, and 17.2R1, only the ACI (suboption 1) and ARI (suboption 2) values from the option 82 information are used. Other suboptions, such as Vendor-Specific (suboption 9), are ignored.

Guidelines for Configuring Support for DHCPv4 Duplicate Clients

When configuring DHCPv4 duplicate client support, consider the following guidelines:

- If you want to preserve the remotely-created option 82 information, use the `option 82` option with the `duplicate-clients-in-subnet` statement to distinguish between duplicate clients. If there is no remotely created option 82 in the incoming DHCP packets, the router locally creates the option 82 information.

- If you want to use the locally-created option-82, use the `incoming-interface` option with the `duplicate-clients-in-subnet` statement to distinguish between duplicate clients.
- Only the ACI (suboption 1) and ARI (suboption 2) values from the option 82 information are used. Other suboptions, such as Vendor-Specific (suboption 9) are ignored.
- DHCP relay agent and DHCP local server in the same routing instance must have the same the `duplicate-clients-in-subnet` configuration.
- For the Layer 3 wholesale model:
 - The wholesaler and retailer logical system/routing instances must have the same `duplicate-clients-in-subnet` statement configuration.
 - For DHCP relay, the wholesaler and the retailer routing contexts must both have the `relay-option-82` statement configured with the Agent Circuit ID suboption (suboption 1) in option 82.

Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Option 82 Information

Duplicate clients occur when two clients in a subnet have the same hardware address or the same client ID.

The following two procedures describe how to configure the router to use the option 82 information in the incoming packets to differentiate between duplicate clients. The first procedure describes the configuration for DHCP relay agent. The second procedure is for DHCP local server.

NOTE: Only the ACI (suboption 1) and ARI (suboption 2) values from the option 82 information are used. Other suboptions, such as Vendor-Specific (suboption 9) are ignored.

To configure the DHCP relay agent to differentiate between duplicate clients based on option 82 information:

1. Specify that you want to configure DHCP relay agent.

```
[edit forwarding-options]  
user@host# edit dhcp-relay
```

2. Configure DHCP relay to insert option 82 information if there is no remotely created option 82. Use the default setting, which inserts the interface ID rather than the optional interface description.

```
[edit forwarding-options dhcp-relay]
user@host# set relay-option-82 circuit-id
```

3. Configure the router to always accept DHCP client packets that contain option 82 information.

```
[edit forwarding-options dhcp-relay]
user@host# set overrides trust-option-82
```

NOTE: The `trust-option-82` statement must always be enabled so the router can process incoming DHCP client packets that contain option 82 information when the packets have a gateway IP address (`giaddr`) of 0 (zero).

4. Configure DHCP relay to use the remotely created option 82 information to distinguish between duplicate clients. If there is no remotely created option 82 in the traffic, the router locally creates the option 82 information.

```
[edit forwarding-options dhcp-relay]
user@host# set duplicate-clients-in-subnet option-82
```

NOTE: Make sure that the `always-write-option-82` statement is *not* enabled, as the statement will overwrite the remotely created option 82.

To configure the DHCP local server to differentiate between duplicate clients based on the option 82 information:

1. Specify that you want to configure DHCP local server.

```
[edit system services]
user@host# edit dhcp-local-server
```

2. Configure the duplicate client support with the `option-82` option.

```
[edit system services dhcp-local-server]
user@host# set duplicate-clients-in-subnet option-82
```

Configuring the Router to Distinguish Between DHCPv4 Duplicate Clients Based on Their Incoming Interfaces

Duplicate clients occur when two clients in a subnet have the same hardware address or the same client ID.

The following two procedures describe how to configure the router to use the clients' incoming interface to differentiate between duplicate clients. The first procedure describes the configuration for DHCP relay agent; the second procedure is for DHCP local server.

To configure the DHCP relay agent to differentiate between duplicate clients based on the client incoming interface:

1. Specify that you want to configure DHCP relay agent.

```
[edit forwarding-options]
user@host# edit dhcp-relay
```

2. Configure the duplicate client support with the `incoming-interface` option.

```
[edit forwarding-options dhcp-relay]
user@host# set duplicate-clients-in-subnet incoming-interface
```

3. Configure DHCP relay to insert option 82 information if the information is not specified remotely. Use the default setting, which inserts the interface ID rather than the optional interface description.

NOTE: Only the ACI (suboption 1) and ARI (suboption 2) values from the option 82 information are used. Other suboptions, such as Vendor-Specific (suboption 9) are ignored.

```
[edit forwarding-options dhcp-relay]
user@host# set relay-option-82 circuit-id
```

4. Configure the router to overwrite any remotely supplied option 82 information in incoming packets.

```
[edit forwarding-options dhcp-relay]
user@host# set overrides always-write-option-82
```

5. Configure the router to always accept DHCP client packets that contain option 82 information.

```
[edit forwarding-options dhcp-relay]
user@host# set overrides trust-option-82
```

NOTE: The *trust-option-82* statement must always be enabled so the router can process incoming DHCP client packets that contain option 82 information when the packets have a gateway IP address (giaddr) of 0 (zero).

To configure the DHCP local server to differentiate between duplicate clients based on the client incoming interface:

1. Specify that you want to configure DHCP local server.

```
[edit system services]
user@host# edit dhcp-local-server
```

2. Configure the duplicate client support with the `incoming-interface` option.

```
[edit system services dhcp-local-server]
user@host# set duplicate-clients-in-subnet incoming-interface
```

Release History Table

Release	Description
16.1R5	Starting in Junos OS Release 16.1R5, 16.2R2, 17.1R2, and 17.2R1, only the ACI (suboption 1) and ARI (suboption 2) values from the option 82 information are used.

RELATED DOCUMENTATION

| [DHCPv6 Duplicate Client Management](#) | 98

DHCPv6 Duplicate Client Management

IN THIS SECTION

- [DHCPv6 Duplicate Client DUIDs | 98](#)
- [Configuring the Router to Use Underlying Interfaces to Distinguish Between DHCPv6 Duplicate Client DUIDs | 99](#)

DHCPv6 Duplicate Client DUIDs

The DHCP unique identifier (DUID) is used to identify a client for the proper application of configuration parameters. The DUID is supposed to be unique across all clients. A duplicate DHCPv6 client occurs when a client attempts to obtain a lease, and that client has the same DUID as an existing DHCPv6 client. Because the DUIDs are supposed to be unique, by default the router treats the request from the duplicate client as a renegotiation by the original client, and replaces the existing client entry with a new entry.

However, in some cases the duplicate request is legitimate, because some network equipment vendors do not guarantee the uniqueness of DUIDs. In these circumstances the router can support the duplication of the DUID by accommodating the new client without affecting the existing client.

Starting in Junos OS Release 16.1, you can enable duplicate DHCPv6 client support. When enabled, the router uses the clients' underlying (incoming) interfaces to differentiate between clients with the same DUID. The router can then create a new client entry for the duplicate client and grant it a lease. The router retains the existing client entry with the original lease.

All underlying interface types are supported. Only 1:1 VLANs are supported, because the client requests are received over different underlying interfaces. N:1 VLANs are not supported, because the client requests can be received over the same underlying interface and therefore cannot be differentiated if the DUIDs are the same.

Configuring the Router to Use Underlying Interfaces to Distinguish Between DHCPv6 Duplicate Client DUIDs

DHCPv6 duplicate clients occur when two clients in a subnet have the same DHCP Unique Identifier (DUID).

The following procedure describes how to configure the router to use the client's underlying (incoming) interface to differentiate between clients with duplicate DUIDs. The first part of the procedure describes the configuration for DHCPv6 relay agent and the second part configures the DHCPv6 local server.

NOTE: Duplicate client DUIDs are supported only when the clients use different underlying interfaces, as in the case of 1:1 VLANs. They are not supported when the clients share an underlying interface, as in the case of N:1 VLANs.

Before configuring duplicate client support, you must ensure the following:

- DHCPv6 relay agent is configured to insert the DHCPv6 Interface-ID option (option 18) in packets forwarded to the DHCPv6 local server.
- Option 18 specifies the interface name, not the text description of the interface.
- DHCPv6 local server must echo option 18 in the RELAY-REPLY messages returned to the DHCPv6 relay agent, as is the case for DHCPv6 local server configured on a Juniper Networks router. The relay agent uses the echoed option 18 information to find the client's interface and construct the client key.

To configure the DHCPv6 relay agent to support duplicate DUIDs:

1. Specify that you want to configure DHCPv6 relay agent.

```
[edit forwarding-options]
user@host# edit dhcp-relay dhcpv6
```

2. Configure DHCPv6 relay agent to insert DHCPv6 option 18 in the packets forwarded to the DHCPv6 local server.

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# set relay-agent-interface-id
```

NOTE: You must not include the `use-interface-description` statement because it specifies a text description of the interface.

3. Specify that the DHCPv6 relay agent uses the clients' incoming interfaces to differentiate between the duplicate DUIDs.

```
[edit forwarding-options dhcp-relay dhcpv6]
user@host# set duplicate-clients incoming-interface
```

To configure the DHCPv6 local server to support duplicate DUIDs:

1. Specify that you want to configure DHCPv6 local server.

```
[edit system services]
user@host# edit dhcp-local-server dhcpv6
```

2. Configure the DHCPv6 local server to support duplicate clients based on the clients' incoming interfaces.

```
[edit system services dhcp-local-server dhcpv6]
user@host# set duplicate-clients incoming-interface
```

Release History Table

Release	Description
16.1	Starting in Junos OS Release 16.1, you can enable duplicate DHCPv6 client support.

RELATED DOCUMENTATION

| [DHCPv4 Duplicate Client Management](#) | 92

3

CHAPTER

PPP Subscriber Access Networks

[PPP Subscriber Access Networks Overview | 102](#)

[PPP Network Control Protocol Negotiation | 126](#)

[Tracing PPP Service Events for Troubleshooting | 136](#)

PPP Subscriber Access Networks Overview

IN THIS SECTION

- [Dynamic Profiles for PPP Subscriber Interfaces Overview | 102](#)
- [Understanding How the Router Processes Subscriber-Initiated PPP Fast Keepalive Requests | 103](#)
- [RADIUS-Sourced Connection Status Updates to CPE Devices | 106](#)
- [Configure Dynamic Profiles for PPP | 111](#)
- [Preventing the Validation of PPP Magic Numbers During PPP Keepalive Exchanges | 112](#)
- [How to Configure RADIUS-Sourced Connection Status Updates to CPE Devices | 113](#)
- [Attaching Dynamic Profiles to Static PPP Subscriber Interfaces | 114](#)
- [Migrating Static PPP Subscriber Configurations to Dynamic Profiles Overview | 115](#)
- [Configuring Local Authentication in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers | 117](#)
- [Configuring Tag2 Attributes in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers | 119](#)
- [Configuring Dynamic Authentication for PPP Subscribers | 120](#)
- [Modifying the CHAP Challenge Length | 122](#)
- [Example: Minimum PPPoE Dynamic Profile | 124](#)
- [Verifying and Managing PPP Configuration for Subscriber Management | 124](#)

Dynamic Profiles for PPP Subscriber Interfaces Overview

Subscriber management PPP support enables you to create and attach dynamic profiles for PPP subscriber interfaces. When the PPP subscriber logs in, the router instantiates the specified dynamic profile and then applies the attributes defined in the profile to the interface.

Dynamic profiles are used for both static and dynamic PPP interfaces. For static PPP interfaces, you use the CLI to attach dynamic profiles, which specify PPP options. For dynamic PPP interfaces, the dynamic profile creates the interface, including the PPP options.

NOTE: Dynamically created interfaces are supported only on PPPoE interfaces.

Unlike traditional PPP support, subscriber management does not allow bi-directional PPP authentication—authentication is performed only by the router, never by the remote peer. The router's AAA process manages authentication and address assignment for subscriber management. When you configure PPP options for a dynamic profile, you can configure either Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) authentication, and you can control the order in which the router negotiates the CHAP and PAP protocols. In addition, for CHAP authentication, you can modify the default length of the CHAP challenge message. Other PPP options, which are either commonly used or mandatory for a traditional PPP interface configuration, are not supported in subscriber management dynamic profiles.

Understanding How the Router Processes Subscriber-Initiated PPP Fast Keepalive Requests

IN THIS SECTION

- [Benefits of PPP Fast Keepalives | 103](#)
- [How PPP Fast Keepalive Processing Works | 104](#)
- [Statistics Display for PPP Fast Keepalive | 104](#)
- [Effect of Changing the Forwarding Class Configuration | 105](#)
- [Ignoring a Magic Number Mismatch | 105](#)

On MX Series routers with Modular Port Concentrators/Modular Interface Cards (MPCs/MICs), the Packet Forwarding Engine on an MPC/MIC processes and responds to Link Control Protocol (LCP) Echo-Request packets that the PPP subscriber (client) initiates and sends to the router. LCP Echo-Request packets and LCP Echo-Reply packets are part of the PPP keepalive mechanism that helps determine whether a link is functioning properly.

Previously, LCP Echo-Request packets and LCP Echo-Reply packets were handled on an MX Series router by the Routing Engine. The mechanism by which LCP Echo-Request packets are processed by the Packet Forwarding Engine instead of by the Routing Engine is referred to as *PPP fast keepalives*.

Benefits of PPP Fast Keepalives

- PPP fast keepalives reduce the time required for keepalive exchanges by enabling the Packet Forwarding Engine to receive LCP Echo-Request packets from the PPP subscriber and respond with

LCP Echo-Reply packets, without having to send the LCP packets to the Routing Engine for processing.

- PPP fast keepalives provide increased bandwidth on the router to support a larger number of subscribers with improved performance by relieving the Routing Engine from having to process the LCP Echo-Request and Echo-Reply packets.
- PPP fast keepalives use negotiated magic numbers to identify potential traffic loopbacks to the router or network issues. You can also disable validation if needed to prevent undesired PPP session termination, for example when the PPP remote peers use arbitrary numbers rather than the negotiated number.

How PPP Fast Keepalive Processing Works

You do not need any special configuration on an MX Series router with MPCs/MICs to enable processing of PPP fast keepalive requests on the Packet Forwarding Engine. The feature is enabled by default, and cannot be disabled.

The following sequence describes how an MX Series router processes LCP Echo-Request packets and LCP Echo-Reply packets on the Packet Forwarding Engine on the MPC/MIC:

1. The Routing Engine notifies the Packet Forwarding Engine when transmission of keepalive requests is enabled on a PPP *logical interface*. The notification includes the magic numbers of both the server and the remote client.
2. The Packet Forwarding Engine receives the LCP Echo-Request packet initiated by the PPP subscriber (client).
3. The Packet Forwarding Engine validates the peer magic number in the LCP Echo-Request packet, and transmits the corresponding LCP Echo-Reply packet containing the magic number negotiated by the router.
4. If the Packet Forwarding Engine detects a loop condition in the link, it sends the LCP Echo-Request packet to the Routing Engine for further processing.

The Routing Engine continues to process LCP Echo-Request packets until the loop condition is cleared.

Transmission of keepalive requests from the Packet Forwarding Engine on the router is not currently enabled.

Statistics Display for PPP Fast Keepalive

When an MX Series router with MPCs/MICs is using PPP fast keepalive for a PPP link, the Keepalive statistics field in the output of the `show interfaces pp0.logical` statistics operational command does not

include statistics for the number of keepalive packets received or sent, or the amount of time since the router received or sent the last keepalive packet.

Effect of Changing the Forwarding Class Configuration

To change the default queue assignment (forwarding class) for outbound traffic generated by the Routing Engine, you can include the `forwarding-class class-name` statement at the `[edit class-of-service host-outbound-traffic]` hierarchy level.

For PPP fast (inline) keepalive LCP Echo-Request and LCP Echo-Reply packets transmitted between an MX Series router with MPCs/MICs and a PPP client, changing the forwarding class configuration takes effect immediately for both new PPP-over-Ethernet (PPPoE), PPP-over-ATM (PPPoA), and L2TP network server (LNS) subscriber sessions created after the configuration change, and for existing PPPoE, PPPoA, and LNS subscriber sessions established before the configuration change.

Ignoring a Magic Number Mismatch

When the Packet Forwarding Engine validates the peer magic number in the received LCP Echo-Request packet, it checks whether the magic number is unexpected. The received number should match the number for the remote peer that was agreed during LCP negotiation. The remote peer number must be different than the local peer number; when they are the same, the expectation is that a loopback condition (traffic is looping back to the local peer) or some other network issue exists.

When the validation check determines that a mismatch is present, meaning that the received remote peer number is different from the negotiated number, the Packet Forwarding Engine sends the failed Echo-Reply packets to the Routing Engine. If an Echo-Reply with a valid magic number is not received within a certain interval, PPP considers this to be a keepalive failure and tears down the PPP session.

Some customer equipment might not negotiate its local magic number and instead insert an arbitrary value as the magic number it sends to the router in the keepalive packets. This number is identified as a mismatch and the session is eventually dropped. Starting in Junos OS Release 18.1R1, this result can be avoided by configuring the router to not perform a magic number validation check. Because the mismatch is never identified, the router continues to exchange PPP keepalive packets with the remote peer. To configure this behavior, include the `ignore-magic-number-mismatch` statement in an L2TP group profile, in the dynamic profile for dynamic PPP subscriber connections terminated at the router, or in the dynamic profile for dynamic tunneled PPP subscribers at the LNS.

SEE ALSO

[Configuring Keepalives](#)

[Disabling the Sending of PPPoE Keepalive Messages](#)

[Changing the Default Queuing and Marking of Host Outbound Traffic](#)

RADIUS-Sourced Connection Status Updates to CPE Devices

IN THIS SECTION

- [Message and Option Formats | 109](#)

Starting in Junos OS Release 20.2R1, you can use RADIUS-sourced messages to convey information that the BNG transparently forwards to a CPE device, such as a home gateway. For example, this information might be upstream bandwidth or some other connection rate parameter that the CPE device needs. This capability is useful when you want to dynamically enforce traffic management as close to subscribers as possible.

Ordinarily, you might use the RADIUS standard attribute Reply-Message (18) to convey this information to the CPE device during PPP authentication. However, if you are already using that attribute for something else, you can also use the Juniper Networks Connection-Status-Message VSA (26-4874–218). This VSA is a logical extension to the Reply-Message attribute (18) and has the same format and semantics.

PPP uses a Juniper Networks vendor-specific extension to LCP to send the contents of the Connection-Status-Message VSA to the peer home gateway. PPP includes this information in the Connection-Status-Message option of an LCP Connection-Update-Request message.

RADIUS can send the Connection-Status-Message VSA to authd in the following ways:

- In the RADIUS Access-Accept message during negotiation and authorization of a PPP session
- In a RADIUS CoA request at any time for an active PPP session

You might use both of these methods for any given session for business or residential subscribers. The Access-Accept message provides the initial connection parameters. The CoA capability enables you to update connection rate parameters as needed throughout the life of a session. The information carried in the Connection-Status-Message VSA is typically traffic rates that are applied by local configuration such as a dynamic service profile or the corresponding ANCP Port Up message.

NOTE: If you do not include the `lcp-connection-update` PPP option in the dynamic client profile, PPP processes the notification from `authd`, but takes no action. If LCP on the router is not in the Opened state, then PPP takes no action on the VSA.

The following steps describe what happens when RADIUS sends the VSA in an Access-Accept message:

1. The `authd` process receives the Connection-Status-Message VSA in an Access-Accept message from the RADIUS server.
2. The `authd` process sends the Connection-Status-Message VSA to PPP (`jpppd`).
3. PPP NCP negotiation takes place between the remote gateway PPP client and PPP on the router.
4. Successful negotiation results in a family activation request. The PPP session enters the Session Up state when the family is activated.
5. If the dynamic client profile includes the `lcp-connection-update` PPP option and LCP on the router is in the Opened state, PPP sends an LCP Connection-Update-Request message to the gateway. This message includes the VSA information in the Connection-Status-Message option.
 - If the gateway supports the LCP Connection-Update-Request, it returns an LCP Connection-Update-Ack message to the router. The home gateway LCP must be in the Opened state when it receives the request, otherwise it discards the request.
 - If the gateway does not support the LCP Connection-Update-Request, it returns an LCP Code-Reject message to the router.

NOTE: If the gateway does not respond, the router retries the update request. It uses the PPP default values of up to a maximum of 10 retries with an interval of 3 seconds between the attempts.

NOTE: If you do not include the `lcp-connection-update` PPP option in the dynamic client profile, PPP processes the notification from `authd`, but takes no action. If the option is present but LCP on the router is not in the Opened state, PPP takes no action regarding the VSA.

The following steps describe what happens when RADIUS sends the VSA in a CoA request. This assumes that NCP negotiation was already successful and the session is active.

1. The `authd` process receives the Connection-Status-Message VSA in a CoA request from the RADIUS server.

2. The authd process sends the Connection-Status-Message VSA to PPP (jpppd).
3. If the dynamic client profile includes the lcp-connection-update PPP option and LCP on the router is in the Opened state, PPP sends an LCP Connection-Update-Request message to the gateway. This message includes the VSA information in the Connection-Status-Message option.
 - If the gateway supports the LCP Connection-Update-Request, it returns an LCP Connection-Update-Ack message to the router. The home gateway LCP must be in the Opened state when it receives the request, otherwise it discards the request.
 - If the gateway does not support the LCP Connection-Update-Request, it returns an LCP Code-Reject message to the router.

NOTE: If the gateway does not respond, the router retries the update request. It uses the PPP default values of up to a maximum of 10 retries with an interval of 3 seconds between the attempts.

If the home gateway fails to receive a Connection-Update-Request message, the router retries sending the message. The router also retries the request when it does not receive either a Connection-Update-Ack or an LCP Code-Reject back from the gateway, or when something is wrong with the Ack message. The default retry interval is 3 seconds. The router will retry the message up to the default 10 times before it quits. If the router exhausts all the retry attempts without receiving an appropriate Connection-Update-Ack message, it logs the message as if it had received a PPP Code-Reject message.

NOTE: RADIUS can include multiple instances of the Connection-Status-Message VSA in either the Access-Accept message or a CoA request. If this occurs, authd uses only the first instance and ignores any others.

The Access-Accept or CoA requests might contain other attributes besides the Connection-Status-Message VSA, but there is no interdependency between the VSA and any other attributes. This is true even when the message includes the Activate-Service (26-65) or Deactivate-Service (26-66) VSAs. The lack of dependency means that even if authd does not successfully apply the other attributes, it still sends the connection info to PPP, which in turn sends the VSA contents to the home gateway.

Similarly, authd applies any other attributes and returns a CoA response regardless of whether PPP successfully communicates the content of the Connection-Status-Message VSA to the remote gateway. This is true even when the CoA contains only the Connection-Status-Message VSA. This capability is necessary because not all gateways will accept the LCP extension used in this feature.

Message and Option Formats

Figure 9 on page 109 shows the format for Connection-Update-Request and Connection-Update-Ack messages. The formats are the same, but Table 6 on page 109 shows that some field values are different for the two messages.

Figure 9: Connection-Update-Request and Connection-Update-Ack Message Format

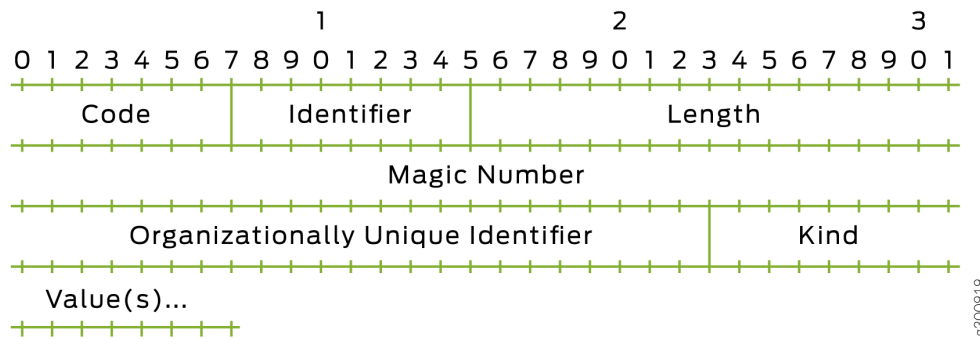


Table 6: Field Values for Connection-Update-Request and Connection-Update-Ack messages

Field	Connection-Update-Request	Connection-Update-Ack
Code	0 for vendor-specific	0 for vendor-specific
Identifier	Identifier for vendor-specific packet	Same identifier as in the Connection-Update-Request message. If this value does not match, the router logs the error and discards the packet. This enables the request message to be retried, just as if the gateway had not received it.
Length	Number of bytes in the packet: 12 plus the length of the Connection-Status-Message option	Number of bytes in the Connection-Update-Ack packet: 12
Magic Number	Negotiated value for the local PPP magic number	Negotiated value for the local PPP magic number

Table 6: Field Values for Connection-Update-Request and Connection-Update-Ack messages
(Continued)

Field	Connection-Update-Request	Connection-Update-Ack
Organizationally Unique Identifier (OUI)	00-21-59 for Juniper Networks	00-21-59 for Juniper Networks
Kind	1 for Session-Update	2 for Session-Ack. For any other value, the router logs the error and the discards the packet. This enables the request message to be retried, just as if the gateway had not received it.
Values	Connection-Status-Message option in TLV format	No values are supported

You can configure how the PPP magic numbers are used.

- If you configure `ignore-magic-number-mismatch` PPP option, you are preventing the magic numbers from being validated. PPP ignores a mismatch between the magic numbers in the request and Ack messages. If there are no other validation errors, PPP accepts the Connection-Update-Ack message.
- If you do not configure `ignore-magic-number-mismatch` PPP option, the magic numbers go through validation. If the magic number in the Ack message does not match the gateway's magic number established during LCP negotiation, the router logs the error and discards the Connection-Update-Ack message as an invalid response. This enables the request message to be retried, just as if the gateway had not received it.

See [Preventing the Validation of PPP Magic Numbers During PPP Keepalive Exchanges](#) for more information about magic numbers.

[Figure 10 on page 111](#) shows the format for the Connection-Status-Message options. [Table 7 on page 111](#) lists the field values.

Figure 10: Connection-Status-Message Option Format

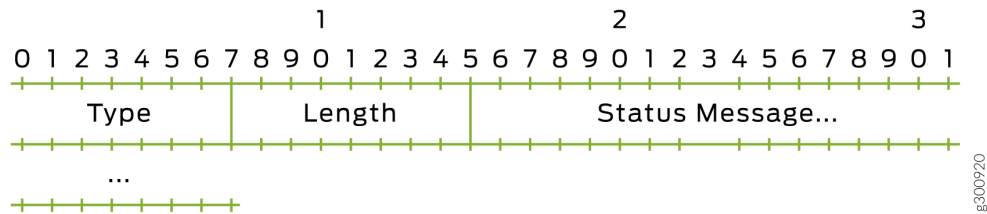


Table 7: Field Values for the Connection-Status-Message Option

Field	Value
Type	1
Length	Number of bytes in the option; 2 plus the length of the message. The message length can be from 1 through 247 bytes.
Status Message	Contents of the Connection-Status-Message VSA

Configure Dynamic Profiles for PPP

A dynamic profile acts as a template that enables you to create, update, or remove a configuration that includes attributes for client access (such as, interface or protocol) or service (such as, IGMP). Using dynamic profiles, you can consolidate all of the common attributes of a client (and eventually a group of clients) and apply the attributes simultaneously.

After dynamic profiles are created, the profiles reside in a profile library on the router. You can then use the `dynamic-profile` statement to attach profiles to interfaces. To assign a dynamic profile to a PPP interface, you can include the `dynamic-profile` statement at the `[edit interfaces interface-name unit logical-unit-number ppp-options]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number ppp-options]
dynamic-profile profile-name;
```

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

For information about dynamic profiles, see *Dynamic Profiles Overview* in the *Junos Subscriber Access Configuration Guide*.

For information about creating dynamic profiles, see *Configuring a Basic Dynamic Profile* in the *Junos Subscriber Access Configuration Guide*.

For information about assigning a dynamic profile to a PPP interface, see ["Attaching Dynamic Profiles to Static PPP Subscriber Interfaces"](#) on page 114 in the *Junos Subscriber Access Configuration Guide*.

For information about using dynamic profiles to authenticate PPP subscribers, see ["Configuring Dynamic Authentication for PPP Subscribers"](#) on page 120.

NOTE: Dynamic profiles for PPP subscribers are supported only on PPPoE interfaces for this release.

Preventing the Validation of PPP Magic Numbers During PPP Keepalive Exchanges

PPP magic numbers are negotiated between peers during LCP negotiation. The peers must have different magic numbers. When the numbers are the same, it indicates that there may be a loopback in traffic sent by the local peer. In this case, the local peer sends a new number to the remote peer. If the magic number returned by the remote peer is different than the latest number sent by the local peer, then the numbers are agreed. Otherwise, the exchange of magic numbers continues until a valid (different) number is received or the process times out, in which case the session is dropped.

After the numbers are agreed upon, the peers include their respective magic numbers when they exchange PPP keepalive (Echo-Request/Echo-Reply) packets. The Packet Forwarding Engine validates the received magic number for each exchange. A mismatch occurs when the PPP magic number received from the remote peer does not match the value agreed upon during LCP negotiation. When the validation check determines that a mismatch is present, the Packet Forwarding Engine sends the failed Echo-Request packet to the Routing Engine. If an Echo-Reply with a valid magic number is not received within a certain interval, PPP considers this to be a keepalive failure and tears down the PPP session.

In some circumstances, this behavior is not desirable. Some customer equipment does not negotiate its local magic number; instead, it inserts an arbitrary value as the magic number it sends to the router in the keepalive packets. By default, this number is identified as a mismatch and the session is eventually dropped. This result can be avoided by preventing the Packet Forwarding Engine from performing the magic number validation check. Because the mismatch is never identified, the router continues to exchange PPP keepalive packets with the remote peer.

Disable the magic number validation check by including the `ignore-magic-number-mismatch` statement as one of the PPP options applied in a dynamic PPP profile, L2TP LNS dynamic profile, or L2TP group profile. Configuring this statement has no effect on LCP magic number negotiation or on the exchange of keepalives when the remote peer magic number is the expected negotiated number.

NOTE: Because magic number validation is not performed, the Packet Forwarding Engine does not detect whether the remote peer sends the local peer's magic number, which would indicate a loopback or other network issue. This is considered to be an unlikely situation, because LCP negotiation completed successfully, meaning no loopback was present at that time.

To configure dynamic profiles to prevent the Packet Forwarding Engine from detecting mismatches in magic numbers:

Configure the PPP option.

- For dynamic PPP subscriber connections terminated at the router:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# set ignore-magic-number-mismatch
```

- For dynamic tunneled PPP subscribers on LNS inline service interfaces:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options]
user@host# set ignore-magic-number-mismatch
```

You can use the `show ppp interface interface-name extensive` command to view whether the magic numbers are ignored.

How to Configure RADIUS-Sourced Connection Status Updates to CPE Devices

You can use RADIUS-sourced messages to convey information that the BNG transparently forwards to a CPE device, such as a home gateway. For example, this information might be upstream bandwidth or some other connection rate parameter that the CPE device needs.

When you enable this feature, PPP can act on a Connection-Status-Message VSA (26–218) received by `authd` in either a RADIUS Access-Accept message or a CoA message. PPP then conveys the contents of

the VSA in an LCP Connection-Update-Request message to the remote peer. This action requires the following to be true:

- At least the first address family has been successfully negotiated and the session is active.
- The router LCP is in the Opened state.

Otherwise PPP takes no action on the VSA. If you do not enable the `lcp-connection-update` option, PPP processes the notification from authd, but takes no action.

You configure this capability in the dynamic client profile associated with subscribers using the CPE device. In practice, you are adding this to numerous other features in the client profile. This example shows only the specific configuration for this feature. This feature also requires you to configure VSA 26-218 on your RADIUS server; that is outside the scope of this documentation.

To configure the connection status updates in a dynamic profile for PPP subscriber interfaces:

1. Edit the PPP options in the client profile.

```
[edit dynamic-profiles ppp-client-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# edit dynamic-profiles ppp-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-options
```

2. Enable the connection status updates.

```
[edit dynamic-profiles ppp-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# set lcp-connection-update
```

You can use the `show ppp interface extensive` command for the PPP logical interface to determine whether LCP connection updates are successful. You can monitor the relevant statistics with the `show system subscriber-management statistics ppp` command.

Attaching Dynamic Profiles to Static PPP Subscriber Interfaces

You can attach a dynamic profile to a static PPP subscriber interface. When a PPP subscriber logs in, the specified dynamic profile is instantiated and the services defined in the profile are applied to the interface.

To attach a dynamic profile to a static PPP subscriber interface:

1. Specify that you want to configure PPP options.

```
[edit interfaces pp0 unit 0]
user@host# edit ppp-options
```

2. Specify the dynamic profile you want to associate with the interface.

```
[edit interfaces pp0 unit 0 ppp-options]
user@host# set dynamic-profile vod-profile-50
```

Migrating Static PPP Subscriber Configurations to Dynamic Profiles

Overview

IN THIS SECTION

- [Local Authentication | 115](#)
- [CPE-Sourced Address Assignment | 116](#)
- [Tag2 Route Attribute | 116](#)
- [Benefits | 117](#)

This topic discusses several considerations for migrating certain static, terminated IPv4 PPP subscriber configurations to dynamic configurations using dynamic profiles. Service providers managing static subscribers on routers with legacy Junos OS releases (earlier than Junos OS Release 15.1R4) have requirements for migrating their static subscribers to being managed with dynamic profiles on routers running enhanced subscriber management (Junos OS Release 15.1R4 and later releases). Starting in Junos OS Release 18.2R1, several enhancements have been added to facilitate the transition of these static service provider configurations to dynamic profiles.

Local Authentication

Some providers with static configurations might use CPE devices that do not support any authentication protocols, not even CHAP or PAP. The providers might use PPPoE service name tables as a rudimentary method to authenticate and authorize the subscribers on static PPPoE logical interfaces. If the subscriber ACI or ARI do not match a table entry, then the PPP PADI and PADR packets are typically

dropped. Legacy Junos OS releases do not support subscribers configured with *no-authentication* for the authentication method.

For subscribers where the CPE does not support authentication protocols such as PAP and CHAP, you can configure usernames and passwords locally. The router uses these values when it contacts the RADIUS server for authentication.

- To configure the username for local authentication, include the `username-include` statement in the PPP options for the dynamic logical interface. You can define the name based on one or more of the following attributes: MAC address, Agent Circuit ID, Agent Remote ID, and domain name. By default, a period (.) is the delimiter between elements of the name, but you can define other characters instead.
- To configure the password for local authentication, include the `password` statement in the PPP options for the dynamic logical interface.

You can use the same dynamic profile to support both CPEs that do not support an authentication protocol and CPEs that do.

CPE-Sourced Address Assignment

For some static configurations, the subscriber address is not assigned by using RADIUS or a local address pool on the router. Instead, the CPE has a static address configured for the subscriber; during IPCP negotiation, the CPE requests the router to assign that address to the subscriber.

Starting in Junos OS Release 18.2R1, you can assign a wildcard address of 255.255.255.255 to the Framed-Route-Address attribute [8] in the configuration for your RADIUS server. When RADIUS returns the attribute with that value, jpppd automatically accepts the subscriber IP address assignment provided by the client in an IPCP configure-request message rather than assigning another address.

Tag2 Route Attribute

In some configurations, static PPP subscriber interfaces are configured in different VRFs. Each VRF configuration has static routes that point to static PPP subscriber interfaces as the next-hop address. These routes might have the tag2 attribute configured; it is required by MP-BGP to apply the appropriate local preference and community when it advertises the routes.

Starting in Junos OS Release 18.2R1, you can configure your RADIUS server to include the tag2 attribute in the Framed-Route attribute [22] when it authenticates a subscriber.

You must also configure the dynamic profile to derive the tag2 value from the Framed-Route attribute. To do so, specify the `$junos-framed-route-tag2` predefined variable to be used when the access route is dynamically instantiated. Alternatively, you can configure the dynamic profile to provide a specific tag2 value for a specific access route prefix.

See *Junos OS Predefined Variables* for more information about predefined variables.

Benefits

- Local authentication enables authentication with locally stored passwords and usernames for subscribers when the CPE does not support authentication protocols such as PAP and CHAP.
- CPE-sourced address assignment enables the router to accept statically configured subscriber IP addresses requested by the CPE rather than assigning the address from a local or externally sourced address pool.
- The tag2 attribute enables more detailed specification of routes.

Configuring Local Authentication in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers

Some providers with static configurations might use CPE devices that do not support any authentication protocols, not even CHAP or PAP. The providers might use PPPoE service name tables as a rudimentary method to authenticate and authorize the subscribers on static PPPoE logical interfaces. If the subscriber ACI or ARI does not match a table entry, then the PPP PADI and PADR packets are typically dropped.

Starting in Junos OS Release 18.2R1, you can configure usernames and passwords locally for clients that do not support authentication protocols such as PAP and CHAP. The router uses these values when it contacts the RADIUS server for authentication. This aids in the migration of the static subscribers to using dynamic profiles on a router running enhanced subscriber management.

To configure local authentication:

1. Configure the username using one or more of the available options.
 - a. (Optional) Specify that the agent circuit identifier (ACI) is included in the username. The ACI is derived from PPPoE tags.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include circuit-id
```

- b. (Optional) Specify that the agent remote ID (ARI) is included in the username. The ARI is derived from PPPoE tags.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include remote-id
```

- c. (Optional) Specify that the MAC address from the client PDU is included in the username. The MAC address is derived from the PPPoE packet.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include mac-address
```

- d. (Optional) Specify the client domain name to end the username. The router adds the @ character as the delimiter for this option.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include domain-name name
```

- e. (Optional) Specify a delimiter to separate the components that make up the username. The default delimiter is a period (.).The router always uses the @ character as the delimiter before the domain name.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include delimiter character
```

2. Configure the password for the subscriber.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set password password
```

The username takes the following format when you include all the options and use the default delimiter:

```
mac-address.circuit-id.remote-id@domain-name
```

For example, consider the following sample configuration, where the ACI is aci1002, the ARI is ari349, and the MAC address is 00:00:5e:00:53:ff:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" ppp-options local-authentication]
user@host# set username-include circuit-id
user@host# set username-include remote-id
user@host# set username-include mac-address
user@host# set username-include domain-name example.com
user@host# set username-include delimiter -
user@host# set password $ABC123$ABC123
```

This configuration results in a local password of \$ABC123\$ABC123 for the following unique local username:

```
0000.5e00.53ff-aci1002-ari349@example.com
```

Configuring Tag2 Attributes in Dynamic Profiles for Static Terminated IPv4 PPP Subscribers

In some configurations, PPP subscribers use static routes with a tag2 attribute. For example, MP-BGP uses tag2 to enable it to apply the appropriate local-preference and community when it advertises routes. When you migrate these subscribers to using dynamic profiles on a router running enhanced subscriber management, you can configure the tag2 attribute by configuring a specific value for a route or by deriving the value from a RADIUS server. This support is first available in Junos OS Release 18.2R1.

- To configure a specific tag2 value for a route:
 - Specify the value.

```
[edit dynamic-profiles profile-name routing-options access route prefix]
user@host# set tag2 route-tag2
```

- To derive the tag2 value from a RADIUS server:

1. Configure your RADIUS server to include the tag2 attribute in the Framed-Route attribute [22] when it authenticates a subscriber. Consult your RADIUS server documentation for configuration information. The configuration might look something like the following example:

```
user@sub.example.com User-Password := "$ABC123"
  Service-Type = Framed-User,
  Framed-Protocol = PPP,
  Framed-Route += "198.51.100.0/24 203.0.113.27 tag 5 distance 10 tag2 3"
```

2. Configure the dynamic profile to use the \$junos-framed-route-tag2 predefined variable to dynamically derive the tag2 value from the Framed-Route attribute.

```
[edit dynamic-profiles profile-name routing-options access route "$junos-framed-route-ip-
address-prefix]
user@host# set tag2 $junos-framed-route-tag2
```

The \$junos-framed-route-ip-address-prefix predefined variable derives the IPv4 address prefix for the access route from the Framed-Route attribute as well.

Configuring Dynamic Authentication for PPP Subscribers

You can configure a dynamic profile that includes PPP authentication that enables PPP clients to dynamically access the network. You can specify either CHAP or PAP authentication. Optionally, you can also control the order in which the router negotiates the CHAP and PAP protocols.

For dynamic interfaces, the router supports unidirectional authentication only—the router always functions as the authenticator. When you configure PPP authentication in a dynamic profile, CHAP authentication supports the challenge-length option, which enables you to configure the minimum length and maximum length of the CHAP challenge message. Neither CHAP authentication nor PAP authentication supports any other configuration options, including the passive statement.

NOTE: Dynamic profiles for PPP subscribers are supported only on PPPoE interfaces.

To configure authentication in a dynamic profile for PPP subscriber interfaces:

1. Name the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles vod-profile-25
```

2. Configure the interfaces and unit for the dynamic profile. Use `pp0` for the interface type and the predefined variable for the unit.

```
[edit dynamic-profiles vod-profile-25]
user@host# edit interfaces pp0 unit $junos-interface-unit
```

3. Configure PPP options.

```
[edit dynamic-profiles vod-profile-25 interfaces pp0 unit "$junos-interface-unit"]
user@host# edit ppp-options
```

4. Specify the authentication protocol used in the dynamic profile. You can configure either CHAP or PAP. There are no additional options for either authentication protocol.

```
[edit dynamic-profiles vod-profile-25 interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# set chap
```

5. (Optional) Configure the minimum length and maximum length of the CHAP challenge message.
See ["Modifying the CHAP Challenge Length" on page 122](#).
6. (Optional) Configure the order in which the router negotiates the CHAP and PAP authentication protocols.
See ["Controlling the Negotiation Order of PPP Authentication Protocols" on page 129](#).
7. (Optional) Configure the router to prompt the CPE to negotiate the DNS addresses for dynamic PPPoE subscribers during IPCP negotiation.

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# set ipcp-suggest-dns-option
```

See ["Ensuring IPCP Negotiation for Primary and Secondary DNS Addresses" on page 134](#) for more information.

Modifying the CHAP Challenge Length

You can modify the default minimum length and maximum length of the Challenge Handshake Authentication Protocol (CHAP) challenge message that the router sends to a PPP client. The CHAP challenge message, which contains information that is unique to a particular PPP subscriber session, is used as part of the authentication mechanism between the router and the client to verify the identity of the client for access to the router.

By default, the minimum length of the CHAP challenge is 16 bytes, and the maximum length is 32 bytes. You can override this default to configure the CHAP challenge minimum length and maximum length in the range 8 bytes through 63 bytes.

BEST PRACTICE: We recommend that you configure both the minimum length and the maximum length of the CHAP challenge to at least 16 bytes.

Before you begin:

- Configure the CHAP protocol on the interface.
 - For dynamic PPP subscriber interfaces, see ["Configuring Dynamic Authentication for PPP Subscribers" on page 120](#).
 - For static interfaces with PPP encapsulation, see *Configuring the PPP Challenge Handshake Authentication Protocol*.

To configure the minimum and maximum length of the CHAP challenge message:

1. Specify that you want to configure PPP options.

- For dynamic PPP subscriber interfaces:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit"]
user@host# edit ppp-options
```

- For static interfaces with PPP encapsulation:

```
[edit interfaces pp0 unit logical-unit-number]
user@host# edit ppp-options
```

2. Specify that you want to configure CHAP options.

- For dynamic PPP subscriber interfaces:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# edit chap
```

- For static interfaces with PPP encapsulation:

```
[edit interfaces pp0 unit logical-unit-number ppp-options]
user@host# edit chap
```

3. Specify the minimum length and maximum length of the CHAP challenge.

- For dynamic PPP subscriber interfaces:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-
options chap]
user@host# set challenge-length minimum minimum-length maximum maximum-length
```

- For static interfaces with PPP encapsulation:

```
[edit interfaces pp0 unit logical-unit-number ppp-options chap]
user@host# set challenge-length minimum minimum-length maximum maximum-length
```

For example, the following challenge-length statement in a dynamic profile named pppoe-client-profile sets the minimum length of the CHAP challenge to 20 bytes, and the maximum length to 40 bytes.

```
[edit dynamic-profiles pppoe-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-
options chap]
user@host# set challenge-length minimum 20 maximum 40
```

Example: Minimum PPPoE Dynamic Profile

This example shows the minimum configuration for a dynamic profile that is used for static PPPoE interfaces. The configuration must include the interfaces `pp0` stanza.

```
dynamic-profiles {
  ppp-profile-1 {
    interfaces {
      pp0 {
        unit "$junos-interface-unit";
      }
    }
  }
}
```

Verifying and Managing PPP Configuration for Subscriber Management

IN THIS SECTION

- [Purpose | 124](#)
- [Action | 124](#)

Purpose

View or clear information about PPP configuration for subscriber management.

Action

- To display information about PPP interfaces:

```
user@host> show ppp interface
```


- To display PPP statistics information:

```
user@host> show ppp statistics
```

- To display PPP session summary information:

```
user@host> show ppp summary
```

- To display PPP address-pool information:

```
user@host>show ppp address-pool
```

Release History Table

Release	Description
20.2R1	Starting in Junos OS Release 20.2R1, you can use RADIUS-sourced messages to convey information that the BNG transparently forwards to a CPE device, such as a home gateway.
18.2R1	Starting in Junos OS Release 18.2R1, several enhancements have been added to facilitate the transition of these static service provider configurations to dynamic profiles.
18.1R1	Starting in Junos OS Release 18.1R1, this result can be avoided by configuring the router to not perform a magic number validation check.

RELATED DOCUMENTATION

[Configuring Keepalives](#)

[Disabling the Sending of PPPoE Keepalive Messages](#)

[Changing the Default Queuing and Marking of Host Outbound Traffic](#)

[Preventing the Validation of PPP Magic Numbers During PPP Keepalive Exchanges | 112](#)

Dynamic Profiles Overview

Configuring a Basic Dynamic Profile

Junos OS Predefined Variables

[Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile | 264](#)

PPP Network Control Protocol Negotiation

IN THIS SECTION

- [PPP Network Control Protocol Negotiation Mode Overview | 126](#)
- [Controlling the Negotiation Order of PPP Authentication Protocols | 129](#)
- [Configuring the PPP Network Control Protocol Negotiation Mode | 132](#)
- [Ensuring IPCP Negotiation for Primary and Secondary DNS Addresses | 134](#)

PPP Network Control Protocol Negotiation Mode Overview

IN THIS SECTION

- [PPP NCP Negotiation Modes | 127](#)
- [PPP NCP Negotiation Mode Supported Configurations | 127](#)
- [PPP NCP Active Negotiation Requirements for IPv4 Dynamic and Static PPP Subscribers | 128](#)
- [PPP NCP Active Negotiation Requirements for IPv6 Dynamic and Static PPP Subscribers | 128](#)
- [PPP NCP Negotiation Requirements for IPv4 and IPv6 Dual-Stack Configurations | 129](#)

The *Network Control Protocol* (NCP) is a mechanism used to establish and configure different Network Layer protocols for Point-to-Point Protocol (PPP) connections. Starting in Junos OS Release 14.1, on MX Series routers with Modular Port Concentrators (MPCs), you can configure *PPP NCP negotiation* to actively or passively control subscriber connections initiated by the router functioning as a PPP server.

Junos OS supports the following NCPs as presented in the associated IETF standards:

- Internet Protocol Control Protocol (IPCP) in RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*
- IPv6 Control Protocol (IPv6CP) in RFC 5072, *IP Version 6 over PPP*

PPP NCP Negotiation Modes

PPP NCP negotiation operates in either of the following modes:

- *Active PPP NCP negotiation mode*—The router sends an NCP Configuration Request message without waiting for the PPP client to do so.
- *Passive PPP NCP negotiation mode*—The router waits for the PPP client to send an NCP Configuration Request message before sending its own Configuration Request message. Dynamic subscriber interface connections and static subscriber interface connections use passive PPP NCP negotiation by default.

Router behavior for active mode and passive mode PPP NCP negotiation differs for dynamic PPP subscribers and static PPP subscribers, as summarized in [Table 8 on page 127](#).

Table 8: PPP NCP Negotiation Mode Behavior for Dynamic and Static Subscribers

PPP Subscribers	PPP NCP Negotiation Mode	Router Behavior
Dynamic	Active	The router establishes the local network address and uses it to send the NCP Configuration Request message without waiting for the PPP client to send a Configuration Request.
Dynamic	Passive	The router establishes the local network address after it receives the NCP Configuration Request message from the PPP client.
Static	Active	The router sends the authentication acknowledgement to the PPP client, and then sends the NCP Configuration Request message without waiting for the PPP client to send its own Configuration Request.
Static	Passive	The router sends the authentication acknowledgement to the PPP client, and then waits for an NCP Configuration Request message from the client before sending a Configuration Request.

PPP NCP Negotiation Mode Supported Configurations

You can configure PPP Network Control Protocol (NCP) negotiation for the following single-stack and dual-stack subscriber configurations on MX Series routers with MPCs:

- Dynamic PPP subscriber connections terminated at the router

- Static PPP subscriber connections terminated at the router
- Dynamic tunneled PPP subscribers at the L2TP network server (LNS)
- Static tunneled PPP subscribers at the L2TP network server (LNS) on an inline service (si) interface

PPP NCP Active Negotiation Requirements for IPv4 Dynamic and Static PPP Subscribers

To configure active PPP IPv4 Network Control Protocol (IPNCP) negotiation for dynamic and static PPP subscribers in a single-stack or dual-stack configuration, make sure you meet the following requirements:

- Configure the IPv4 (inet) protocol family in a dynamic profile (for dynamic subscribers) or at the interface level (for static subscribers).
- Assign any of the following IPv4 address attributes for the subscriber during the authentication process:
 - Framed-IP-Address (RADIUS Attribute 8)—RADIUS explicit IPv4 address
 - Framed-Pool (RADIUS Attribute 88)—RADIUS IPv4 address pool name
 - IPv4 attributes allocated from a locally configured address pool

When you have met these requirements, use the `initiate-ncp ip` statement to enable active IPNCP negotiation for dynamic and static subscribers in a single-stack or dual-stack configuration.

PPP NCP Active Negotiation Requirements for IPv6 Dynamic and Static PPP Subscribers

To configure active PPP IPv6 Network Control Protocol (IPv6NCP) negotiation for dynamic and static PPP subscribers in a single-stack or dual-stack configuration, make sure you meet the following requirements:

- Configure the IPv6 (inet6) protocol family in a dynamic profile (for dynamic subscribers) or at the interface level (for static subscriber).
- Assign any of the following IPv6 address attributes for the subscriber during the authentication process:
 - Delegated-IPv6-Prefix (RADIUS Attribute 123)—RADIUS explicit IPv6 address
 - Framed-IPv6-Prefix (RADIUS Attribute 97)—RADIUS explicit IPv6 prefix
 - Framed-IPv6-Pool (RADIUS Attribute 100)—RADIUS explicit IPv6 address or prefix pool name

- IPv6 attributes allocated from a locally configured Neighbor Discovery Router Advertisement (NDRA) pool

When you have met these requirements, use the `initiate-ncp ipv6` statement to enable active IPv6NCP negotiation for dynamic and static subscribers in a single-stack or dual-stack configuration.

PPP NCP Negotiation Requirements for IPv4 and IPv6 Dual-Stack Configurations

You can configure either active or passive PPP NCP negotiation for the IPv4 and IPv6 subscriber interfaces in a dual-stack configuration.

To configure active negotiation in a dual-stack configuration, do all of the following:

- Make sure you meet the IPv4 and IPv6 protocol and address family requirements.
- Use the `initiate-ncp ip` statement to enable active negotiation for the IPv4 subscriber interface.
- Use the `initiate-ncp ipv6` statement to enable active negotiation for the IPv6 subscriber interface.

To configure passive negotiation in a dual-stack configuration, do both of the following:

- Make sure you meet the IPv4 and IPv6 protocol and address family requirements.
- Use the `initiate-ncp dual-stack-passive` statement to enable passive negotiation for the dual-stack configuration. The `initiate-ncp dual-stack-passive` statement overrides the `initiate-ncp ip` and `initiate-ncp ipv6` statements if they are configured.

The following additional guidelines apply when you configure PPP NCP negotiation for dual-stack subscribers:

- Dual-stack subscribers configured for either active mode or passive mode PPP NCP negotiation continue to use the same negotiation mode when the NCP mechanism is renegotiated.
- Using the `on-demand-ip-address` statement to save IPv4 addresses for dual-stack PPP subscribers when you are not using the IPv4 service has no effect on configuration of the PPP NCP negotiation mode in a dual-stack configuration.

Controlling the Negotiation Order of PPP Authentication Protocols

You can control the order in which the router tries to negotiate PPP authentication protocols when it verifies that a PPP client can access the network. By default, the router first tries to negotiate Challenge Handshake Authentication Protocol (CHAP) authentication. If the attempt to negotiate CHAP authentication is unsuccessful, the router then tries to negotiate Password Authentication Protocol (PAP) authentication.

You can modify this default negotiation order in any of the following ways:

- Specify that the router negotiate PAP authentication first, followed by CHAP authentication if PAP negotiation is unsuccessful.

When you specify both authentication protocols in either order, you must enclose the set of protocol names in square brackets ([]).

- Specify that the router negotiate only CHAP authentication.
- Specify that the router negotiate only PAP authentication.

Before you begin:

- Configure the CHAP or PAP protocol on the interface.
 - For dynamic PPP subscriber interfaces, see ["Configuring Dynamic Authentication for PPP Subscribers" on page 120](#).
 - For CHAP on static interfaces with PPP encapsulation, see *Configuring the PPP Challenge Handshake Authentication Protocol*.
 - For PAP on static interfaces with PPP encapsulation, see *Configuring the PPP Password Authentication Protocol On a Physical Interface*.
 - For information about dynamic profiles for PPP subscribers, see ["Dynamic Profiles for PPP Subscriber Interfaces Overview" on page 102](#).

To control the order in which the router negotiates PPP authentication protocols:

1. Specify that you want to configure PPP options.

- For dynamic PPP subscriber interfaces:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit"]
user@host# edit ppp-options
```

- For static interfaces with PPP encapsulation:

```
[edit interfaces pp0 unit logical-unit-number]
user@host# edit ppp-options
```

2. Specify the negotiation order for PPP authentication protocols on the router.

- For dynamic PPP subscriber interfaces:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# set authentication [authentication-protocols]
```

- For static interfaces with PPP encapsulation:

```
[edit interfaces pp0 unit logical-unit-number ppp-options]
user@host# set authentication [authentication-protocols]
```

The following sample authentication statements in a dynamic profile named pppoe-client-profile show the different ways you can configure the negotiation order for PPP authentication protocols. (The authentication statements for configuring static interfaces are identical.)

- To specify that the router negotiate PAP authentication first, followed by CHAP authentication:

```
[edit dynamic-profiles pppoe-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# set authentication [pap chap]
```

- To specify that the router negotiate only CHAP authentication:

```
[edit dynamic-profiles pppoe-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# set authentication chap
```

- To specify that the router negotiate only PAP authentication:

```
[edit dynamic-profiles pppoe-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# set authentication pap
```

- To restore the default negotiation order for PPP authentication protocols after you have modified it:

```
[edit dynamic-profiles pppoe-client-profile interfaces pp0 unit "$junos-interface-unit" ppp-
options]
user@host# set authentication [chap pap]
```

Configuring the PPP Network Control Protocol Negotiation Mode

Starting in Junos OS Release 14.1, configuring PPP Network Control Protocol (NCP) negotiation enables you to actively or passively control subscriber connections initiated by the router functioning as a PPP server. Both dynamic and static subscriber interface connections use passive PPP NCP negotiation by default.

You can configure the PPP NCP negotiation mode (active or passive) for the following subscriber configurations on MX Series routers with MPCs:

- Dynamic PPP subscriber connections terminated at the router, using a dynamic profile
- Static PPP subscriber connections terminated at the router, using a per-interface configuration
- Dynamic tunneled PPP subscribers at the L2TP network server (LNS), using a dynamic profile
- Static tunneled PPP subscribers at the LNS, using a per-inline service (si) interface configuration
- Dynamic and static tunneled PPP subscribers at the LNS, using a user-group profile

To configure PPP NCP negotiation mode:

1. Specify that you want to configure PPP-specific properties for the subscriber.

- For dynamic PPP subscriber connections terminated at the router:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit"]
user@host# edit ppp-options
```

- For static PPP subscriber connections terminated at the router:

```
[edit interfaces pp0 unit logical-unit-number]
user@host# edit ppp-options
```


- For dynamic tunneled PPP subscribers at the LNS:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# edit ppp-options
```

- For static tunneled PPP subscribers at the LNS:

```
[edit interfaces si-fpc/pic/port unit logical-unit-number]
user@host# edit ppp-options
```

- In a group profile for dynamic and static tunneled PPP subscribers at the LNS:

```
[edit access group-profile profile-name ppp]
user@host# edit ppp-options
```

2. Configure PPP NCP negotiation mode in any of the following ways:

- To configure active PPP NCP negotiation for IPv4 subscribers in a single-stack or dual-stack configuration, use the `initiate-ncp ip` statement.

For example, to configure active negotiation for static IPv4 connections terminated at the router:

```
[edit interfaces pp0 unit logical-unit-number ppp-options]
user@host# initiate-ncp ip
```

- To configure active PPP NCP negotiation for IPv6 subscribers in a single-stack or dual-stack configuration, use the `initiate-ncp ipv6` statement.

For example, to configure active negotiation for dynamic IPv6 connections terminated at the router:

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# initiate-ncp ipv6
```

- To configure passive PPP NCP negotiation for dynamic or static subscribers in an IPv4 and IPv6 dual-stack configuration, use the `initiate-ncp dual-stack-passive` statement, which overrides both the `initiate-ncp ip` and `initiate-ncp ipv6` statements if they are configured.

For example, to configure passive negotiation for dynamic tunneled PPP subscribers at the LNS in an IPv4 and IPv6 dual-stack configuration:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# initiate-ncp dual-stack-passive
```

Ensuring IPCP Negotiation for Primary and Secondary DNS Addresses

Starting in Junos OS Release 15.1, you can configure a router to prompt any customer premises equipment (CPE) to send the IPv4 primary or secondary DNS address options in the next configuration request if the options are not included in an initial IPCP configuration request during IPCP negotiations or if the router rejects the request. This DNS option enables the router to control IPv4 DNS address provisioning for dynamic and static, terminated PPPoE and LNS subscribers. The router includes the address options in the IPCP configuration NAK message that it sends to the CPE. The CPE then negotiates both primary and secondary IPv4 DNS addresses. Using this option ensures that the CPE can use the DNS addresses available at the router.

To configure the router to prompt the CPE to negotiate the DNS addresses for dynamic PPPoE subscribers:

- Specify the DNS negotiation option.

```
[edit dynamic-profiles profile-name interfaces pp0 unit "$junos-interface-unit" ppp-options]
user@host# set ipcp-suggest-dns-option
```

To configure the router to prompt the CPE to negotiate the DNS addresses for static PPPoE subscribers:

- Specify the DNS negotiation option.

```
[edit interfaces interface-name ppp-options]
user@host# set ipcp-suggest-dns-option
```

To configure the router to prompt the CPE to negotiate the DNS addresses for dynamic LNS subscribers:

- Specify the DNS negotiation option.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit
"          $junos-interface-unit"          ppp-options]
user@host# set ipcp-suggest-dns-option
```

To configure the router to prompt the CPE to negotiate the DNS addresses for static LNS subscribers:

- Specify the DNS negotiation option.

```
[edit interfaces si-slot/pic/port unit logical-unit-number ppp-options]
user@host# set ipcp-suggest-dns-option
```

To configure the router to prompt the CPE to negotiate the DNS addresses for tunneled PPP subscribers with an LNS user group profile:

- Specify the DNS negotiation option.

```
[edit access group-profile profile-name ppp-options]
user@host# set ipcp-suggest-dns-option
```

Release History Table

Release	Description
15.1	Starting in Junos OS Release 15.1, you can configure a router to prompt any customer premises equipment (CPE) to send the IPv4 primary or secondary DNS address options in the next configuration request if the options are not included in an initial IPCP configuration request during IPCP negotiations or if the router rejects the request.
14.1	Starting in Junos OS Release 14.1, on MX Series routers with Modular Port Concentrators (MPCs), you can configure <i>PPP NCP negotiation</i> to actively or passively control subscriber connections initiated by the router functioning as a PPP server.
14.1	Starting in Junos OS Release 14.1, configuring PPP Network Control Protocol (NCP) negotiation enables you to actively or passively control subscriber connections initiated by the router functioning as a PPP server.

RELATED DOCUMENTATION

[Configuring the PPP Attributes for a Group Profile](#)

[Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile | 264](#)

[Dynamic Profiles Overview](#)

[Configuring Dynamic Authentication for PPP Subscribers | 120](#)

Tracing PPP Service Events for Troubleshooting

IN THIS SECTION

- [Configuring the PPP Service Trace Log Filename | 137](#)
- [Configuring the Number and Size of PPP Service Log Files | 138](#)
- [Configuring Access to the PPP Service Log File | 138](#)
- [Configuring a Regular Expression for PPP Service Messages to Be Logged | 139](#)
- [Configuring Subscriber Filtering for PPP Service Trace Operations | 139](#)
- [Configuring the PPP Service Tracing Flags | 141](#)
- [Configuring the Severity Level to Filter Which PPP Service Messages Are Logged | 141](#)

The Junos OS trace feature tracks PPP service operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `jpppd`. You can specify a different filename, but you cannot change the directory in which trace files are located.
2. When the trace log file `filename` reaches 128 kilobytes (KB), it is compressed and renamed `filename.0.gz`. Subsequent events are logged in a new file called `filename`, until it reaches capacity again. At this point, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the [System Log Explorer](#).)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

To configure PPP service tracing operations:

1. (Optional) Configure a trace log filename.

See "[Configuring the PPP Service Trace Log Filename](#)" on page 137.

2. (Optional) Configure the number and size of trace logs.

See "[Configuring the Number and Size of PPP Service Log Files](#)" on page 138.

3. (Optional) Configure user access to trace logs.

See "[Configuring Access to the PPP Service Log File](#)" on page 138.

4. (Optional) Configure a regular expression to filter the information to be included in the trace log.

See "[Configuring a Regular Expression for PPP Service Messages to Be Logged](#)" on page 139.

5. (Optional) Configure flags to specify which events are logged.

See "[Configuring the PPP Service Tracing Flags](#)" on page 141.

6. (Optional) Configure a severity level for messages to specify which event messages are logged.

See "[Configuring the Severity Level to Filter Which PPP Service Messages Are Logged](#)" on page 141.

Configuring the PPP Service Trace Log Filename

By default, the name of the file that records trace output for PPP service is `jpppd`. You can specify a different name with the `file` option.

To configure the filename for PPP service tracing operations:

- Specify the name of the file used for the trace output.

```
[edit protocols ppp-service traceoptions]
user@host# set file ppp-service_logfile_1
```

Configuring the Number and Size of PPP Service Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format `.number.gz`. The newest archived file is `.0.gz` and the oldest archived file is `.(maximum number)-1.gz`. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, `filename`, reaches 2 MB, `filename` is compressed and renamed `filename.0.gz`, and a new file called `filename` is created. When the new `filename` reaches 2 MB, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until there are 20 trace files. Then the oldest file, `filename.19.gz`, is simply overwritten when the next oldest file, `filename.18.gz` is compressed and renamed to `filename.19.gz`.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit protocols ppp-service traceoptions]
user@host# set file ppp-service_1 _logfile_1 files 20 size 2097152
```

Configuring Access to the PPP Service Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit protocols ppp-service traceoptions]
user@host# set file ppp-service_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit protocols ppp-service traceoptions]
user@host# set file ppp-service_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for PPP Service Messages to Be Logged

By default, the trace operation output includes all lines relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit protocols ppp-service traceoptions]
user@host# set file ppp-service_1 _logfile_1 match regex
```

Configuring Subscriber Filtering for PPP Service Trace Operations

You can apply filters to the PPP service to limit tracing to particular subscribers or domains. Subscriber filtering simplifies troubleshooting in a scaled environment by enabling you to focus on a reduced set of trace results.

For subscriber usernames that have the expected form of *user@domain*, you can filter on the user, the domain, or both. You can use an asterisk (*) as a wildcard to substitute for characters at the beginning or end of either term or both terms to match a greater number of subscribers.

NOTE: You cannot filter results using a wildcard in the middle of the user or domain terms. For example, the following uses of the wildcard are not supported: *tom*25@example.com*, *tom125@ex*.com*.

When you enable filtering by username, traces that have insufficient information to determine the username are automatically excluded.

To configure subscriber filtering:

- Specify the filter.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user user@domain
```

Consider the following examples of using the wildcard for filtering:

- Filter results for the specific subscriber with the username, tom@example.com.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user tom@example.com
```

- Filter results for all subscribers whose username begins with tom.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user tom*
```

- Filter results for all subscribers whose username ends with tom.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user *tom
```

- Filter results for subscribers with the username tom at all domains beginning with ex.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user tom@ex*
```

- Filter results for all subscribers at all domains that end with ample.com.

```
[edit protocols ppp-service traceoptions]
user@host# set filter user *ample.com
```


- Filter results for all subscribers whose username begins with tom at domains that end with example.com.

```
[edit protocols ppp-service traceoptions]  
user@host# set filter user tom*@*example.com
```

Configuring the PPP Service Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit protocols ppp-service traceoptions]  
user@host# set flag flag
```

Configuring the Severity Level to Filter Which PPP Service Messages Are Logged

The messages associated with a logged event are categorized according to severity level. You can use the severity level to determine which messages are logged for the event type. A low severity level is less restrictive—filters out fewer messages—than a higher level. When you configure a severity level, all messages at that level and all higher (more restrictive) levels are logged.

The following list presents severity levels in order from lowest (least restrictive) to highest (most restrictive). This order also represents the significance of the messages; for example, error messages are of greater concern than info messages.

- verbose
- info
- notice
- warning

- error

The severity level that you configure depends on the issue that you are trying to resolve. In some cases you might be interested in seeing all messages relevant to the logged event, so you specify `all`. You can also specify `verbose` with the same result, because `verbose` is the lowest (least restrictive) severity level; it has nothing to do with the terseness or verbosity of the messages. Either choice generates a large amount of output. You can specify a more restrictive severity level, such as `notice` or `info` to filter the messages. By default, the trace operation output includes only messages with a severity level of `error`.

To configure the type of messages to be logged:

- Configure the message severity level.

```
[edit protocols ppp-service traceoptions]  
user@host# set level severity
```

RELATED DOCUMENTATION

| [PPP Subscriber Access Networks Overview](#) | 102

4

CHAPTER

L2TP Subscriber Access Networks

[L2TP for Subscriber Access Overview | 144](#)

[L2TP Tunnel Switching For Multiple-Domain Networks | 158](#)

[L2TP LAC Subscriber Configuration | 176](#)

[L2TP LAC Tunneling for Subscribers | 182](#)

[L2TP Subscriber Access Lines and Connection Speeds | 219](#)

[L2TP LNS Inline Service Interfaces | 259](#)

[IP Packet Reassembly on Inline Service Interfaces | 317](#)

[Peer Resynchronization After an L2TP Failover | 322](#)

[Tracing L2TP Events for Troubleshooting | 326](#)

L2TP for Subscriber Access Overview

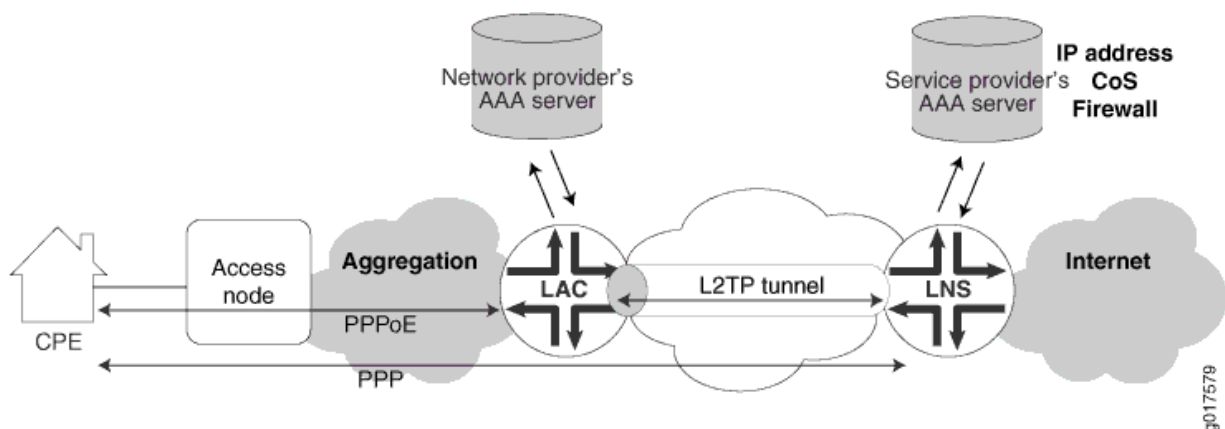
IN THIS SECTION

- L2TP for Subscriber Access Overview | 144
- L2TP Terminology | 147
- L2TP Implementation | 148
- Retransmission of L2TP Control Messages | 151
- Configuring Retransmission Attributes for L2TP Control Messages | 152
- Enabling Tunnel and Global Counters for SNMP Statistics Collection | 154
- Verifying and Managing L2TP for Subscriber Access | 155

L2TP for Subscriber Access Overview

The Layer 2 Tunneling Protocol (L2TP) is a client-server protocol that allows the Point-to-Point Protocol (PPP) to be tunneled across a network. L2TP encapsulates Layer 2 packets, such as PPP, for transmission across a network. An L2TP access concentrator (LAC), configured on an access device, receives packets from a remote client and forwards them to an L2TP network server (LNS) on a remote network. The LNS functions as the logical termination point of the PPP session tunneled by the LAC from the remote client. [Figure 11 on page 144](#) shows a simple L2TP topology.

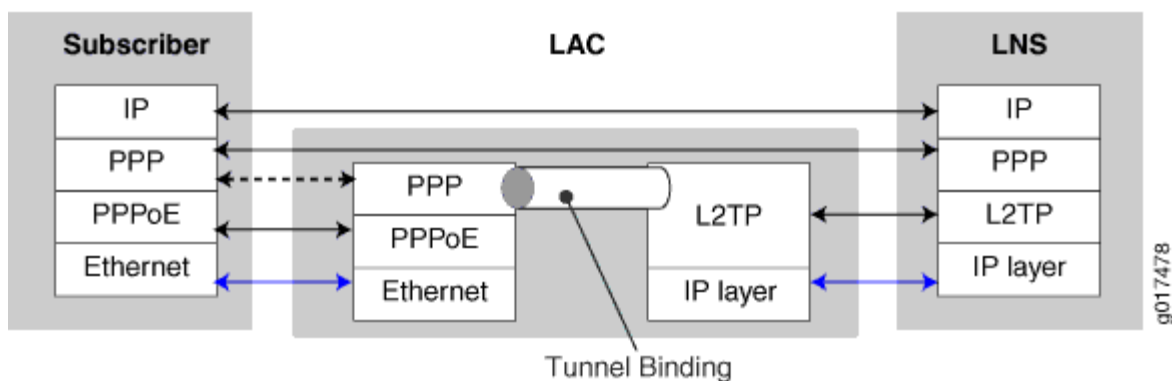
Figure 11: Typical L2TP Topology



L2TP separates the termination of access technologies, such as cable or xDSL, from the termination of PPP and subsequent access to a network. This separation enables public ISPs to outsource their access technologies to competitive local exchange carriers (CLECs). L2TP provides ISPs the capability to supply VPN service; private enterprises can reduce or avoid investment in access technologies for remote workers.

You can configure your router to act as the LAC in PPP pass-through mode in which the LAC receives packets from a remote client and then forwards them at Layer 2 directly to the LNS. The PPP session is terminated on the LNS. This LAC implementation supports only Point-to-Point Protocol over Ethernet (PPPoE) subscribers over dynamic or static logical interfaces. [Figure 12 on page 145](#) shows the protocol layer stacking for an L2TP pass-through connection.

Figure 12: Protocol Stacking for L2TP Subscribers in Pass-Through Mode



NOTE: On MX Series routers, the LAC and LNS functions are supported only on MPCs; they are not supported on any services PIC or MS-DPC. For details about MPC support for L2TP, see the [MX Series Interface Module Reference](#)

Certain M Series routers support LNS functions on services PICs. For more information about the L2TP implementation on M Series routers, see [L2TP Services Configuration Overview](#).

The LAC dynamically creates tunnels based on AAA authentication parameters and transmits L2TP packets to the LNS by means of the IP/User Datagram Protocol (UDP). Traffic travels in an L2TP *session*; a tunnel is an aggregation of one or more sessions. You can also provision a domain map that is used by AAA to determine whether to tunnel or terminate the PPPoE subscriber on the LAC. A one-to-one mapping exists between each PPP subscriber tunneled to the LNS and an L2TP session.

When the LNS is an MX Series router, a LAC-facing peer interface on an MPC provides an IP address for the exchange of IP packets between the tunnel endpoints; the Routing Engine maintains the L2TP tunnels. The Packet Forwarding Engine hosts one or more inline services (*si*) interfaces. These interfaces function like a virtual physical interface and *anchor* the L2TP sessions on the LNS. The *si* interface

enables L2TP services without requiring a special services PIC. Finally, another interface is used to transmit the subscriber data to and from the Internet.

The characteristics of the tunnel can originate either from a tunnel profile that you configure or from RADIUS tunnel attributes and vendor-specific attributes (VSAs) from the AAA server accessible at the LAC. You can include a tunnel profile in a domain map, which applies the tunnel profile before RADIUS authentication takes place. You can use RADIUS standard attributes and VSAs to override any or all characteristics configured by the tunnel profile in a domain map. Alternatively, RADIUS can itself apply a tunnel profile when the RADIUS Tunnel-Group VSA [26-64] is specified in the RADIUS login.

NOTE: L2TP is not supported over GRE tunnels.

The Virtual-Router VSA [26-1] in the subscriber profile on the service provider AAA server (accessible from the LNS) determines the routing instance in which the L2TP session is brought up on the LNS. When this VSA is not present, the subscriber session comes up in the same routing instance as the tunnel, because the AAA server can be accessed only from the routing instance in which the tunnel terminates on the LNS.

This behavior is different than for DHCP and non-tunneled PPPoE subscribers, which come up in the default routing instance in the absence of the Virtual-Router VSA. For L2TP subscribers, you must include this VSA in the subscriber profile when you want the subscriber session to come up in a different routing instance than the tunnel routing instance.

Starting in Junos OS Release 17.4R1, The LNS includes the following RADIUS attributes when it sends an Access-Request message to the RADIUS server:

- Tunnel-Type (64)
- Tunnel-Medium-Type (65)
- Tunnel-Client-Endpoint (66)
- Tunnel-Server-Endpoint (67)
- Acct-Tunnel-Connection (68)
- Tunnel-Assignment-Id (82)
- Tunnel-Client-Auth-Id (90)
- Tunnel-Server-Auth-Id (91)

In earlier releases, the LNS includes those attributes only in the accounting records it sends to the RADIUS server. In the Access-Request messages, they can be used to correlate on the RADIUS server the session from the LAC to the LNS.

The LAC supports RADIUS-initiated mirroring, which creates secure policies based on certain RADIUS VSAs, and uses RADIUS attributes to identify a subscriber whose traffic is to be mirrored. (This feature is not supported for an LNS configured on an MX Series router.)

The LAC and the LNS support unified ISSU. When an upgrade is initiated, the LAC completes any L2TP negotiations that are in progress but rejects any new negotiations until the upgrade has completed. No new tunnels or sessions are established during the upgrade. Subscriber logouts are recorded during the upgrade and are completed after the upgrade has completed.

L2TP Terminology

Table 9 on page 147 describes the basic terms for L2TP.

Table 9: L2TP Terms

Term	Description
AVP	Attribute value pair (AVP)—Combination of a unique attribute—represented by an integer—and a value containing the actual value identified by the attribute.
Call	A connection (or attempted connection) between a remote system and the LAC.
LAC	L2TP access concentrator (LAC)—A node that acts as one side of an L2TP tunnel endpoint and is a peer to the LNS. The LAC sits between an LNS and a remote system and forwards packets to and from each.
LNS	L2TP network server (LNS)—A node that acts as one side of an L2TP tunnel endpoint and is a peer to the LAC. The LNS is the logical termination point of a PPP connection that is being tunneled from the remote system by the LAC.
Peer	In the L2TP context, either the LAC or LNS. The LAC's peer is an LNS, and vice versa.
Proxy authentication	PPP pre-authentication performed by the LAC on behalf of the LNS. The proxy data is sent by the LAC to the LNS containing attributes such as authentication type, authentication name, and authentication challenge. The LNS responds with the authentication results.

Table 9: L2TP Terms (Continued)

Term	Description
Proxy LCP	Link Control Protocol (LCP) negotiation that is performed by the LAC on behalf of the LNS. The proxy is sent by the LAC to the LNS containing attributes such as the last configuration attributes sent and received from the client.
Remote system	An end system or router attached to a remote access network, which is either the initiator or recipient of a call.
Session	A logical connection created between the LAC and the LNS when an end-to-end PPP connection is established between a remote system and the LNS. NOTE: There is a one-to-one relationship between established L2TP sessions and their associated PPP connections.
Tunnel	A connection between the LAC-LNS pair consisting of a control connection and 0 or more L2TP sessions.

L2TP Implementation

IN THIS SECTION

- [Sequence of Events on the LAC | 149](#)
- [Sequence of Events on the LNS | 150](#)

L2TP is implemented on four levels:

- Source—The local router acting as the LAC.
- Destination—The remote router acting as the LNS.
- Tunnel—A direct path between the LAC and the LNS.
- Session—A PPP connection in a tunnel.

When the router has established destinations, tunnels, and sessions, you can control the L2TP traffic. Making a change to a destination affects all tunnels and sessions to that destination; making a change to a tunnel affects all sessions in that tunnel. For example, closing a destination closes all tunnels and sessions to that destination.

Sequence of Events on the LAC

The router acting as the LAC creates destinations, tunnels, and sessions dynamically, as follows:

1. The client initiates a PPP connection with the router.
2. The router and the client exchange Link Control Protocol (LCP) packets. The LAC negotiates on behalf of the LNS; this is known as *proxy LCP*.
3. The LAC authenticates the client on behalf of the LNS; this is known as *proxy authentication*. By using either a local database related to the domain name or RADIUS authentication, the router determines either to terminate or to tunnel the PPP connection.
4. If the router discovers that it should tunnel the session, it does the following:
 - a. Sets up a new destination or selects an existing destination.
 - b. Sets up a new tunnel or selects an existing tunnel.

When a shared secret is configured in either the tunnel profile or the RADIUS attribute Tunnel-Password [69]—depending on which method is used to configure the tunnel—the secret is used to authenticate the tunnel during the establishment phase. The LAC includes the Challenge AVP in the SCCRQ message sent to the LNS. The LNS returns the Challenge Response AVP in the SCCRQ message. If the response from the LNS does not match the value expected by the LAC, then tunnel authentication fails and the tunnel is not established.

- c. Opens a new session.
5. The router forwards the results of the LCP negotiations and authentication to the LNS.

A PPP connection now exists between the client and the LNS.

NOTE: The router discards received packets if the size of the variable-length, optional offset pad field in the L2TP header is too large. The router always supports packets that have an offset pad field of up to 16 bytes, and may support larger offset pad fields, depending on other information in the header. This restriction is a possible, although unlikely, cause of excessive discarding of L2TP packets.

NOTE: When the LAC terminates a PPP session, it generates a PPP disconnect cause and includes this information in the PPP Disconnect Cause Code (AVP 46) when it sends a Call-Disconnect-Notify (CDN) message to the LNS. The code value is 0, which indicates a global error with no information available.

Sequence of Events on the LNS

A router acting as an LNS might be set up as follows:

1. The LAC initiates a tunnel with the router acting as the LNS.
2. The LNS verifies that a tunnel with this LAC is valid: the destination is configured, the hostname and the tunnel password are correct.
3. The LNS completes the tunnel setup with the LAC.
4. The LAC sets up a session and initiates a session request to the LNS.
5. The LNS uses a static interface or creates a dynamic interface to anchor the PPP session.
6. If they are enabled and present, the LNS accepts the proxy LCP and the proxy authentication data and passes them to PPP.
7. PPP processes the proxy LCP, if it is present, and, if the proxy LCP is acceptable, places LCP on the LNS in opened state without renegotiation of LCP.
8. PPP processes the proxy authentication data, if it is present, and passes the data to AAA for verification. (If the data is not present, PPP requests the data from the peer.)

NOTE: When the proxy LCP is not present or not acceptable, the LNS negotiates LCP with the peer. When LCP renegotiation is enabled on the LNS, the LNS ignores any pre-negotiated LCP parameters and renegotiates both the LCP parameters and PPP authentication with the PPP client.

9. The LNS passes the authentication results to the peer.

Retransmission of L2TP Control Messages

L2TP peers maintain a queue of control messages that must be sent to the peer device. After the local peer (LAC or LNS) sends a message, it waits for a response from the remote peer. If a response is not received, the local peer retransmits the message. This behavior allows the remote peer more time to respond to the message.

You can control the retransmission behavior in the following two ways:

- **Retransmission count**—You can configure how many times an unacknowledged message is retransmitted by the local peer. Increasing the count provides more opportunities for the remote peer to respond, but also increases the amount of control traffic. For tunnels that have been established, include the `retransmission-count-established` statement at the `[edit services l2tp tunnel]` hierarchy level. For tunnels that are not yet established, include the `retransmission-count-not-established` statement.
- **Retransmission interval**—You can configure how long the local peer waits for the first response to a control message. If a response is not received within the first timeout interval, then the retransmission timer doubles the interval between each successive retransmission up to a maximum of 16 seconds. Increasing the interval gives the remote peer more time to respond, but also spends more resources on a potentially unavailable peer. Include the `minimum-retransmission-interval` statement at the `[edit services l2tp tunnel]` hierarchy level.

The local peer continues retransmitting the control message until one of the following occurs:

- A response is received within the current waiting period.
- The maximum retransmission count is reached.

If the maximum count is reached and no response has been received, the tunnel and all its sessions are cleared.

NOTE: Reaching the maximum interval of 16 seconds does not halt retransmissions. The local peer continues to wait 16 seconds after each subsequent retransmission.

The following examples describe the retransmission behavior in different circumstances:

- **Example 1**—The retransmission count is three and the minimum retransmission interval is 1 second.
 1. The local peer sends a control message.
 2. The local peer waits 1 second, but receives no response.
 3. The local peer retransmits the control message. This is the first retransmission.

4. The local peer waits 2 seconds, but receives a response before the interval expires.
 5. Retransmission stops because a response is received within the interval.
- Example 2—The retransmission count is two and the minimum retransmission interval is 8 seconds.
 1. The local peer sends a control message.
 2. The local peer waits 8 seconds, but receives no response.
 3. The local peer retransmits the control message. This is the first retransmission.
 4. The local peer waits 16 seconds, but receives no response.
 5. The local peer retransmits the control message. This is the second retransmission.
 6. The local peer again waits 16 seconds, because the interval cannot increase beyond 16, but receives no response.
 7. Retransmission stops because the maximum retransmission count of two was reached.
 8. The tunnel and all its sessions are cleared.

Configuring Retransmission Attributes for L2TP Control Messages

You can control the retransmission of unacknowledged L2TP control messages by configuring how many times the local peer retransmits the message and how long it waits for a response before retransmission.

L2TP peers maintain a queue of control messages that must be sent to the peer device. After the local peer (LAC or LNS) sends a message, it waits for a response from the remote peer. If a response is not received within the minimum retransmission interval, the local peer retransmits the message and waits for double the retransmission interval. Each time it retransmits the message, the peer doubles how long it waits, up to a maximum of 16 seconds.

If no response is received, the local peer continues to send the message until the number of retransmissions matches the retransmission count. In this case, retransmissions stop and the tunnel and all its sessions are cleared.

BEST PRACTICE: Before you downgrade to a Junos OS Release that does not support these statements, we recommend that you explicitly unconfigure the feature by including the `no retransmission-count-established` statement and the `no retransmission-count-non-established` statement at the `[edit services l2tp tunnel]` hierarchy level.

BEST PRACTICE: During a unified in-service software upgrade (unified ISSU) on an MX Series router configured as the LAC, the LAC does not respond to control messages from the LNS. This can result in dropping LAC L2TP sessions. You can avoid this situation by ensuring that the maximum retransmission count on the LNS is set to 16 or higher.

To set the maximum retransmission count for established tunnels:

- Configure the count.

```
[edit services l2tp tunnel]
user@host# set retransmission-count-established count
```

To set the maximum retransmission count for non-established tunnels:

- Configure the count.

```
[edit services l2tp tunnel]
user@host# set retransmission-count-not-established count
```

To set the minimum interval between retransmissions:

- Configure the interval.

```
[edit services l2tp tunnel]
user@host# set minimum-retransmission-timeout seconds
```

For example, the following configuration specifies that established tunnels have a maximum retransmission count of three and a minimum retransmission interval of two seconds:

```
[edit services l2tp tunnel]
user@host# set retransmission-count-established 3
user@host# set minimum-retransmission-timeout 2
```

With this sample configuration, the following sequence applies to each control message sent by the LAC or LNS:

1. The local peer sends the control message and waits for a response from the remote peer.

2. If the response is not received within the minimum interval of 2 seconds, the local peer retransmits the message. This is the first retransmission.
3. If the response is not received within 4 seconds, the local peer retransmits the message. This is the second retransmission.
4. If the response is not received within 8 seconds, the local peer retransmits the message. This is the third and final retransmission, because the maximum count has been reached.
5. If the response is not received within 16 seconds, the tunnel and all its sessions are cleared.

Enabling Tunnel and Global Counters for SNMP Statistics Collection

By default, SNMP polling is disabled for L2TP statistics. As a consequence, the L2TP tunnel and global counters listed in [Table 10 on page 154](#) have a default value of zero.

Table 10: SNMP Counters for L2TP Statistics

Counter Name	Type
jnxL2tpTunnelStatsDataTxPkts	Tunnel
jnxL2tpTunnelStatsDataRxPkts	Tunnel
jnxL2tpTunnelStatsDataTxBytes	tunnel
jnxL2tpTunnelStatsDataRxBytes	Tunnel
jnxL2tpStatsPayloadRxOctets	Global
jnxL2tpStatsPayloadRxPkts	Global
jnxL2tpStatsPayloadTxOctets	Global
jnxL2tpStatsPayloadTxPkts	Global

You can enable collection of these statistics by including the `enable-snmp-tunnel-statistics` statement at the `[edit services l2tp]` hierarchy level. When enabled, the L2TP process polls for these statistics every 30 seconds for 1000 sessions. The potential age of the statistics increases with the number of subscriber

sessions; the data is refreshed more quickly as the number of sessions decreases. For example, with 60,000 sessions, none of these statistics can be more than 30 minutes old.

BEST PRACTICE: The system load can increase when you enable these counters and also use RADIUS interim accounting updates. We recommend you enable these counters when you are using only SNMP statistics.

To enable L2TP statistics collection for SNMP:

- Enable statistics collection.

```
[edit services l2tp]
user@host1# set enable-snmp-tunnel-statistics
```

Verifying and Managing L2TP for Subscriber Access

IN THIS SECTION

- Purpose | 155
- Action | 156

Purpose

View or clear information about L2TP tunnels and sessions.

BEST PRACTICE: The all option is not intended to be used as a means to perform a bulk logout of L2TP subscribers. We recommend that you do not use the all option with the clear services l2tp destination, clear services l2tp session, or clear services l2tp tunnel statements in a production environment. Instead of clearing all subscribers at once, consider clearing subscribers in smaller group, based on interface, tunnel, or destination end point.

Action

- To display a summary of L2TP tunnels, sessions, errors, and control and data packets:

```
user@host> show services l2tp summary
```

- To display the L2TP destinations:

```
user@host> show services l2tp destination
```

- To clear all L2TP destinations:

```
user@host> clear services l2tp destination all
```

- To clear statistics for all L2TP tunnels belonging to a destination, tunnels belonging to a specified local-gateway address, and tunnels belonging to a specified peer-gateway address:

```
user@host>clear services l2tp destination statistics all  
user@host>clear services l2tp destination local-gateway 203.0.113.2
```

- To display the L2TP sessions:

```
user@host> show services l2tp session
```

- To clear all L2TP sessions, the session with a specified local session ID, or sessions associated with the local gateway specified by an IP address or name:

```
user@host>clear services l2tp session all  
user@host>clear services l2tp session local-session-id 40553  
user@host>clear services l2tp session local-gateway 203.0.113.2  
user@host>clear services l2tp session local-gateway-name lns-mx960
```


- To clear statistics for all L2TP sessions, the session with a specified local session ID, or sessions associated with the local gateway specified by an IP address or name:

```
user@host>clear services l2tp session statistics all
user@host>clear services l2tp session statistics local-session-id 17967
user@host>clear services l2tp session statistics local-gateway 203.0.113.2
user@host>clear services l2tp session statistics local-gateway-name lns-mx960
```

- To display the L2TP tunnels:

```
user@host> show services l2tp tunnel
```

- To clear all L2TP tunnels, the tunnel with a specified local tunnel ID, or tunnels associated with the local gateway specified by an IP address or name:

```
user@host> clear services l2tp tunnel all
user@host>clear services l2tp tunnel local-tunnel-id 40553
user@host>clear services l2tp tunnel local-gateway 203.0.113.2
user@host>clear services l2tp tunnel local-gateway-name lns-mx960
```

- To clear statistics for all L2TP tunnels, the tunnel with a specified local tunnel ID, or tunnels associated with the local gateway specified by an IP address or name:

```
user@host> clear services l2tp tunnel statistics all
user@host>clear services l2tp tunnel statistics local-tunnel-id 40553
user@host>clear services l2tp tunnel statistics local-gateway 203.0.113.2
user@host>clear services l2tp tunnel statistics local-gateway-name lns-mx960
```

RELATED DOCUMENTATION

[Configuring an L2TP LAC | 176](#)

[Configuring an L2TP LNS with Inline Service Interfaces | 259](#)

RADIUS IETF Attributes Supported by the AAA Service Framework

Juniper Networks VSAs Supported by the AAA Service Framework

[Configuring a Tunnel Profile for Subscriber Access | 210](#)

Domain Mapping Overview

L2TP Tunnel Switching For Multiple-Domain Networks

IN THIS SECTION

- [L2TP Tunnel Switching Overview | 158](#)
- [Tunnel Switching Actions for L2TP AVPs at the Switching Boundary | 163](#)
- [Configuring L2TP Tunnel Switching | 168](#)
- [Setting the L2TP Receive Window Size | 171](#)
- [Setting the L2TP Tunnel Idle Timeout | 171](#)
- [Setting the L2TP Destruct Timeout | 172](#)
- [Configuring the L2TP Destination Lockout Timeout | 172](#)
- [Removing an L2TP Destination from the Destination Lockout List | 173](#)
- [Configuring L2TP Drain | 174](#)
- [Using the Same L2TP Tunnel for Injection and Duplication of IP Packets | 175](#)

L2TP Tunnel Switching Overview

IN THIS SECTION

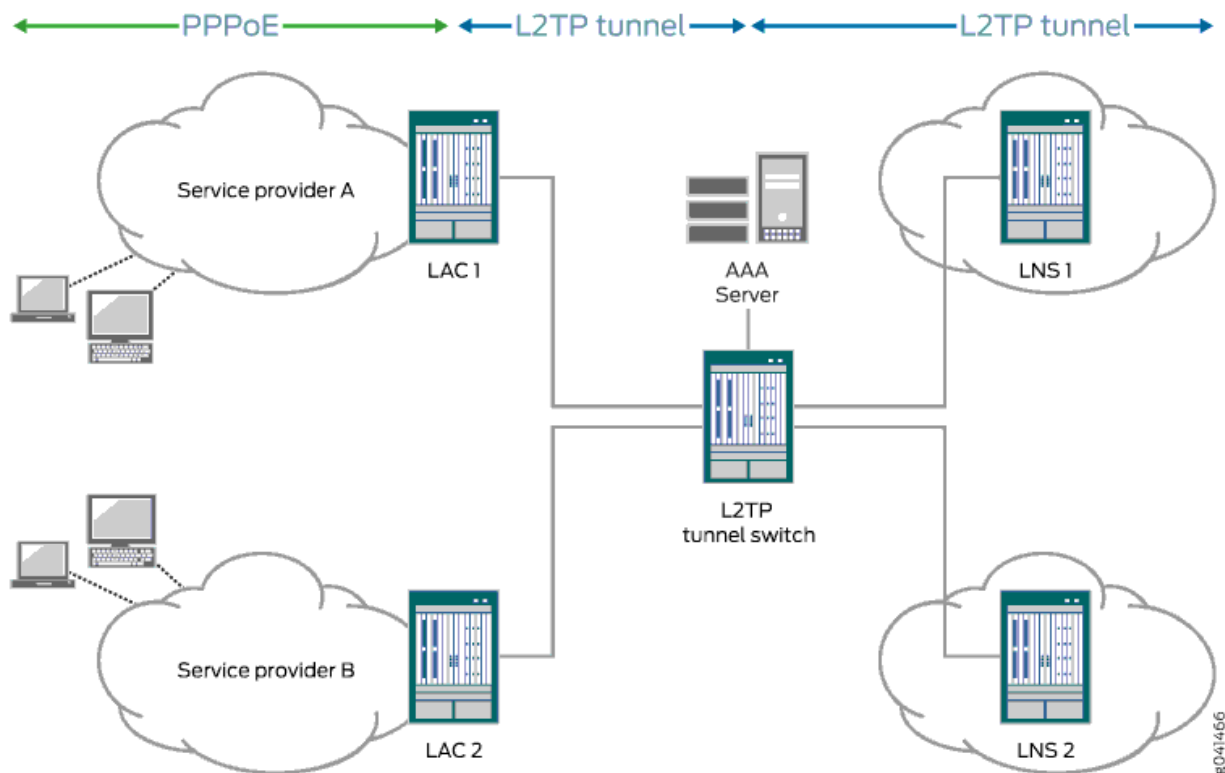
- [Application of Tunnel Switch Profiles | 160](#)
- [Termination of Tunnel-Switched Sessions on the LTS | 161](#)

L2TP tunnel switching, also known as L2TP multihop, simplifies the deployment of an L2TP network across multiple domains. A router that lies between a LAC and an LNS is configured as an *L2TP tunnel switch* (LTS)—sometimes referred to simply as a *tunnel switch* or a *tunnel switching aggregator* (TSA)—as shown in [Figure 13 on page 159](#). The LTS is configured as both an LNS and a LAC. When a remote LAC sends encapsulated PPP packets to the LNS configured on the LTS, the LTS can forward or redirect the

packets through a different tunnel to a different LNS beyond the LTS. The logical termination point of the original L2TP session is switched to a different endpoint.

For example, in the network shown in [Figure 13 on page 159](#), packets from the subscriber provisioned by service provider A are initially targeted at the LNS configured on the LTS. The LTS might redirect those packets to LNS1.

Figure 13: L2TP Tunnel Switching Network Topology



L2TP tunnel switching simplifies network configuration when the administrative domain of a LAC is different from that of the desired LNS. For example:

- The LTS acts as the LNS for multiple LACs. The individual LACs do not have to have the administrative control or capability required to identify the most appropriate LNS on which to terminate their sessions. The LTS performs that function is centralized in the LTS.
- The LTS acts as the LAC for multiple LNSs. When a new remote LAC is added to an ISP's network, the ISP does not have to reconfigure its LNS routers to accommodate the new LAC, because they connect to the LAC on the LTS.

In a Layer 2 wholesale network, the wholesaler can use L2TP tunnel switching to create a flatter network configuration that is easier to manage. The wholesaler bundles Layer 2 sessions from a LAC

that are destined for different ISPs—and therefore different LNSs—onto a single L2TP tunnel. This configuration enables a common L2TP control connection to be used for the LAC.

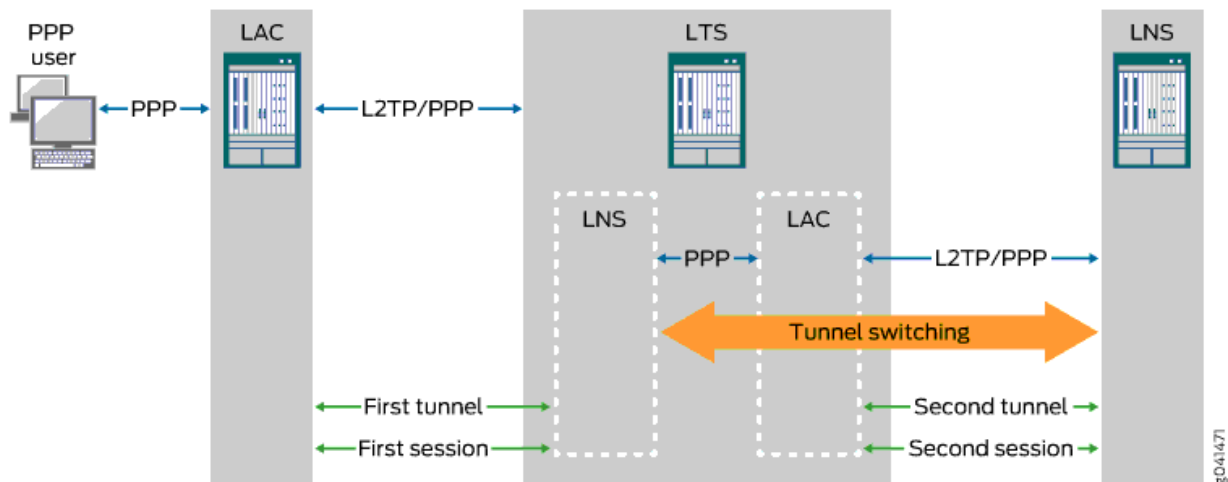
Figure 14 on page 160 shows an example of L2TP tunnel switching for incoming calls with the following sequence of events:

1. The subscriber opens a PPP session to the LAC.
2. The LAC creates the first L2TP tunnel to the LNS configured on the LTS and the first L2TP session to carry the encapsulated PPP packets.
3. During authentication of this first session, the LTS determines whether to retunnel the session to an LNS beyond the LTS, based on the presence or absence of a tunnel switch profile configured on the LTS.

The tunnel switch profile can be a default profile or it can be applied by the RADIUS server, a domain map configuration, or a tunnel group configuration.

4. If a tunnel switch profile is configured, the LTS creates a second tunnel (if it does not already exist) to the LNS beyond the LTS as specified in the profile and creates the second session in this tunnel.

Figure 14: L2TP Tunnel Switching for Incoming Calls



Application of Tunnel Switch Profiles

You can configure a tunnel switch profile to be applied in several ways:

- As a default profile applied globally to traffic received from all LACs
- With a domain map applied to a subscriber session

- With a tunnel group applied to a subscriber session
- In your RADIUS server configuration, returned in the Tunnel Switch-Profile VSA (26-91)

You can configure more than one of these methods of application. When multiple tunnel switch profiles are present, the following order of precedence establishes which profile the LTS uses; the order is from highest (RADIUS) to lowest (default profile):

1. RADIUS VSA 26-91 > domain map > tunnel group > global tunnel switch profile

The tunnel switch profile must also reference a tunnel profile. This tunnel profile specifies the characteristics of the second tunnel, to which the subscriber packets are switched.

Termination of Tunnel-Switched Sessions on the LTS

Tunnel switched sessions are terminated on the LTS when any of the following happens:

- Either the LAC or LNS interface on the LTS receives a Call-Disconnect-Notify (CDN) message ([Table 11 on page 161](#)).

Table 11: Cause of CDN Message

CDN Message Is Received On	When
LAC interface	Either of the following occurs: <ul style="list-style-type: none"> • The second session cannot be established. • The remote LNS terminates the second session.
LNS interface	Either of the following occurs: <ul style="list-style-type: none"> • The PPPoE client initiates a logout. • The originating LAC initiates termination of the tunnel

Both the first and second sessions are terminated because the LTS relays the CDN to the interface that did not receive the CDN. The disconnect cause is the same for both sessions.

- Either the LAC or LNS interface on the LTS receives a Stop-Control-Connection-Notification (StopCCN) message ([Table 12 on page 162](#)).

Table 12: Cause of StopCCN Message

StopCCN Message Is Received On	When
LAC interface	Either of the following occurs: <ul style="list-style-type: none"> • The second session cannot be established. • The remote LNS terminates the second tunnel.
LNS interface	The originating LAC initiates termination of the tunnel.

The LTS does not relay the StopCCN message, because a given tunnel can contain both switched and nonswitched sessions. Another reason in a wholesale scenario is that the tunnel ending on the LNS on the LTS can contain sessions from LACs from different providers. Instead, the LTS sends a CDN message to the interface that did not receive the StopCCN to terminate the tunnel-switched session. This CDN relays the error code carried in the StopCCN.

- An administrative clear command is issued on the LTS.

[Table 13 on page 162](#) lists the actions taken when an administrative clear command is issued on the LTS.

Table 13: LAC, LNS, and LTS Actions Taken for Switched Tunnels in Response to Administrative clear Commands

Command	LAC or LNS Action	LTS Action
clear services l2tp destination	Clear the destination and all associated tunnels and sessions.	For each switched session in a tunnel to the destination, clear the corresponding mapped switched session by sending it a CDN message with the cause set to Administrative.
clear services l2tp destination all	Clear all destinations and all associated tunnels and sessions.	None.
clear services l2tp session	Clear the session.	Clear the corresponding mapped switched session for this session by sending it a CDN message with the cause set to Administrative.

Table 13: LAC, LNS, and LTS Actions Taken for Switched Tunnels in Response to Administrative clear Commands (Continued)

Command	LAC or LNS Action	LTS Action
clear services l2tp session all	Clear all sessions.	None.
clear services l2tp tunnel	Clear the tunnel and all its sessions.	For each switched session in the tunnel, clear the corresponding mapped switched session by sending it a CDN message with the cause set to Administrative.
clear services l2tp tunnel all	Clear all tunnels.	None.

Tunnel Switching Actions for L2TP AVPs at the Switching Boundary

When L2TP tunnel switching redirects packets to a different LNS, it performs one of the following default actions at the switching boundary for each AVP carried in the L2TP messages:

- **relay**—L2TP transparently forwards the AVP in the switched packet with no alteration.
- **regenerate**—L2TP ignores the received AVP that was negotiated by the first tunnel and session. It generates a new AVP for the second session based on the local policy at the LTS and sends this AVP in the switched packet. The local policy may or may not use the value for the AVP received during negotiation for the first session.

[Table 14 on page 164](#) lists the default action for each AVP. Mandatory AVPs are always included in the L2TP messages from the LAC; optional AVPs might be included in the messages.

You can optionally override the default action taken at the switching boundary for the Bearer Type AVP (18), Calling Number AVP (22), or Cisco NAS Port Info AVP (100). You can configure any of these three AVPs to be dropped from the switched packets or regenerated, or you can restore the default relay action.

NOTE: L2TP AVPs that have their attribute values hidden are always regenerated at the switching boundary. The value is decoded and sent in clear text when the packet is forwarded to the remote LNS.

Table 14: Default Action for Handling L2TP AVPs at the Switching Boundary

AVP Name (Number)	AVP Type	L2TP Message Type	Default Action
Assigned Session Id (14)	Mandatory	CDN, ICRQ	Regenerate
Assigned Tunnel Id (9)	Mandatory	SCCRQ	Regenerate
Bearer Capabilities (4)	Optional	SCCRQ	Regenerate
Bearer Type (18)	Optional	ICRQ	Relay
Call Serial Number (15)	Mandatory	ICRQ	Relay
Called Number (21)	Optional	ICRQ	Relay
Calling Number (22)	Optional	ICRQ	Relay
Challenge (11)	Optional	SCCRQ	Regenerate
Challenge Response (13)	Optional	SCCCN	Regenerate
Cisco NAS Port	Optional	ICRQ	Relay
Failover Capability	Optional	SCCRQ	Regenerate
Firmware Revision (6)	Optional	SCCRQ	Regenerate
Framing Capabilities (3)	Mandatory	SCCRQ	Regenerate

Table 14: Default Action for Handling L2TP AVPs at the Switching Boundary (Continued)

AVP Name (Number)	AVP Type	L2TP Message Type	Default Action
Framing Type (19)	Mandatory	ICCN	Relay
Host Name (7)	Mandatory	SCCRQ	Regenerate
Initial Received LCP CONFREQ (26)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, the AVP is regenerated rather than relayed.
Last Received LCP CONFREQ (28)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, the AVP is regenerated rather than relayed.
Last Sent LCP CONFREQ (27)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, the AVP is regenerated rather than relayed.
Message Type (0)	Mandatory	All	Regenerate
Physical Channel Id (25)	Optional	ICRQ	Regenerate

Table 14: Default Action for Handling L2TP AVPs at the Switching Boundary (Continued)

AVP Name (Number)	AVP Type	L2TP Message Type	Default Action
Private Group Id (37)	Optional	ICCN	Relay
Protocol Version (2)	Mandatory	SCCRQ	Regenerate
Proxy Authen Challenge (31)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, authentication is also renegotiated and the AVP is regenerated rather than relayed.
Proxy Authen ID (32)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, authentication is also renegotiated and the AVP is regenerated rather than relayed.
Proxy Authen Name (30)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, authentication is also renegotiated and the AVP is regenerated rather than relayed.

Table 14: Default Action for Handling L2TP AVPs at the Switching Boundary (Continued)

AVP Name (Number)	AVP Type	L2TP Message Type	Default Action
Proxy Authen Response (33)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, authentication is also renegotiated and the AVP is regenerated rather than relayed.
Proxy Authen Type (29)	Optional	ICCN	Relay When LCP renegotiation is enabled with the lcp-negotiation statement in the client profile on the LNS, authentication is also renegotiated and the AVP is regenerated rather than relayed.
Receive Window Size (10)	Optional	SCCRQ	Regenerate
Rx Connect Speed (38)	Optional	ICCN	Relay
Sequencing Required (39)	Optional	ICCN	Regenerate
Sub-Address (23)	Optional	ICRQ	Relay
Tie Breaker (5)	Optional	SCCRQ	Regenerate
Tunnel Recovery	Optional	SCCRQ	Regenerate
Tx Connect Speed (24)	Mandatory	ICCN	Relay

Table 14: Default Action for Handling L2TP AVPs at the Switching Boundary (Continued)

AVP Name (Number)	AVP Type	L2TP Message Type	Default Action
Vendor Name (8)	Optional	SCCRQ	Regenerate

Configuring L2TP Tunnel Switching

L2TP tunnel switching enables a router configured as an LTS to forward PPP packets carried on one L2TP session to a second L2TP session terminated on a different LNS. To configure L2TP tunnel switching, you must define a tunnel switch profile and then assign that profile.

You can configure tunnel switch profiles for all sessions globally, all sessions in a tunnel group, all sessions in a domain or in your RADIUS server configuration to be returned in the RADIUS Tunnel Switch-Profile VSA (26-91). The order of precedence for tunnel switch profiles from various sources is as follows:

- RADIUS VSA 26-91 > domain map > tunnel group > global tunnel switch profile

To define an L2TP tunnel switch profile:

1. Create the profile.

```
[edit access]
user@host# edit tunnel-switch-profile profile-name
```

2. (Optional) Override the default actions taken for certain L2TP AVPs at the switching boundary.

```
[edit access tunnel-switch-profile profile-name]
user@host# set avp bearer-type action
user@host# set avp calling-number action
user@host# set avp cisco-nas-port-info action
```

3. Specify the tunnel profile that defines the tunnel to which the subscriber traffic is switched.

NOTE: This step is not required for a tunnel switch profile specified in the Tunnel Switch-Profile VSA (26-91).

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

4. (Optional) Apply the profile as a global default profile to switch packets from all incoming sessions from the LAC.

```
[edit services l2tp]
user@host1# set tunnel-switch-profile profile-name
```

5. (Optional) Apply the profile as part of a tunnel group to switch packets from all sessions in the tunnel group.

```
[edit services l2tp tunnel-group name]
user@host1# set tunnel-switch-profile profile-name
```

NOTE: The tunnel group is part of the LTS configuration that enables it to act as the LNS for the original sessions from the LAC.

A tunnel group with a tunnel switch profile must also contain a dynamic profile, because tunnel switching supports only dynamic subscribers.

6. (Optional) Apply the profile as part of a domain map to switch packets from all sessions that are associated with the domain.

```
[edit access domain map domain-map-name]
user@host1# set tunnel-switch-profile profile-name
```

NOTE: A domain map cannot have both a tunnel switch profile and a tunnel profile. You must remove one if you add the other.

7. (Optional) Apply the profile by means of the Tunnel-Switch-Profile VSA [26-91] in the RADIUS Access-Accept message returned when the session from the LAC is authenticated. Refer to the documentation for your RADIUS server to determine how to configure this method.

NOTE: A tunnel switch profile specified by a RADIUS server in the Tunnel Switch-Profile VSA (26-91) takes precedence over the tunnel switch profile specified in the CLI configuration. If the Tunnel-Group VSA (26-64) is received in addition to the Tunnel Switch-Profile VSA (26-91), the Tunnel Switch-Profile VSA (26-91) takes precedence over the Tunnel-Group VSA (26-64), ensuring that the subscribers are tunnel switched rather than LAC tunneled.

For example, consider the following configuration, which creates three tunnel switch profiles, l2tp-tunnel-switch-profile, lts-profile-groupA, and lts-profile-example-com:

```
[edit access tunnel-switch-profile l2tp-tunnel-switch-profile]
user@host# set avp bearer-type regenerate
user@host# set avp calling-number regenerate
user@host# set avp cisco-nas-port-info drop
user@host# set tunnel-profile l2tp-tunnel-profile1

[edit access tunnel-switch-profile lts-profile-groupA]
user@host# set tunnel-profile l2tp-tunnel-profile2
[edit access tunnel-switch-profile lts-profile-example.com]
user@host# set tunnel-profile l2tp-tunnel-profile3

[edit services l2tp]
user@host1# set tunnel-switch-profile l2tp-tunnel-switch-profile
user@host1# set tunnel-group groupA tunnel-switch-profile lts-profile-groupA

[edit access domain]
user@host1# set map example.com tunnel-switch-profile lts-profile-example.com
```

The profile l2tp-tunnel-switch-profile is applied as the global default. When packets are switched according to this profile, the values for the Bearer Type AVP (18) and Calling Number AVP (22) in the L2TP packets are regenerated based on local policy at the L2TP tunnel switch and then sent with the packets. The Cisco NAS Port Info AVP (100) is simply dropped. Finally, l2tp-tunnel-profile1 provides the configuration characteristics of the tunnel to which the traffic is switched.

Tunnel switch profile lts-profile-groupA is applied by means of a tunnel group, groupA; it specifies a different tunnel profile, l2tp-tunnel-profile2 and it does not override any AVP actions. Tunnel switch profile lts-profile-example.com is applied by means of a domain map for the example.com domain; it specifies a different tunnel profile, l2tp-tunnel-profile3 and it does not override any AVP actions.

Setting the L2TP Receive Window Size

You can configure the L2TP receive window size for an L2TP tunnel. The receive window size specifies the number of packets a peer can send before waiting for an acknowledgment from the router.

By default, the receive window size is set to four packets. If the receive window size is set to its default value, the router does not send the Receive Window Size AVP, AVP 10, in its first packet sent during tunnel negotiation to its peer.

To configure the receive window size:

```
[edit services l2tp tunnel]
user@host# set rx-window-size packets
```

Setting the L2TP Tunnel Idle Timeout

You can configure the LAC or the LNS to specify how long a tunnel without any sessions remains active. The idle timer starts when the last session on the tunnel is terminated. When the timer expires the tunnel is disconnected. This idle timeout frees up resources otherwise consumed by inactive tunnels.

If you set the idle timeout value to zero, the tunnel is forced to remain active indefinitely after the last session is terminated until one of the following occurs:

- You issue the `clear services l2tp tunnel` command.
- The remote peer disconnects the tunnel.

BEST PRACTICE: Before you downgrade to a Junos OS Release that does not support this statement, we recommend that you explicitly unconfigure the feature by including the `no idle-timeout` statement at the `[edit services l2tp tunnel]` hierarchy level.

To set the tunnel idle timeout:

- Configure the timeout period.

```
[edit services l2tp tunnel]
user@host# set idle-timeout seconds
```

Setting the L2TP Destruct Timeout

You can configure the LAC or the LNS to specify how long the router attempts to maintain dynamic destinations, tunnels, and sessions after they have been destroyed. This destruct timeout aids debugging and other analysis by saving underlying memory structures after the destination, tunnel, or session is terminated. Any specific dynamic destination, tunnel, or session may not be maintained for this entire time period if the resources must be reclaimed early to allow new tunnels to be established.

BEST PRACTICE: Before you downgrade to a Junos OS Release that does not support this statement, we recommend that you explicitly unconfigure the feature by including the `no destruct-timeout` statement at the `[edit services l2tp]` hierarchy level.

To set the L2TP destruct timeout:

- Configure the timeout period.

```
[edit services l2tp]
user@host# set destruct-timeout seconds
```

Configuring the L2TP Destination Lockout Timeout

When multiple sets of tunneling parameters are available, L2TP uses a selection process to choose the best tunnel for subscriber traffic. As part of this selection process, L2TP locks out destinations it cannot connect to when a subscriber tries to reach a domain. L2TP places the destination on the destination lockout list and excludes the destination from consideration for a configurable period called the *destination lockout timeout*.

By default, the destination lockout timeout is 300 seconds (5 minutes). You can configure a value from 60 through 3600 seconds (1 minute through 1 hour). When the lockout timeout expires, L2TP assumes that the destination is now available and includes the destination when performing the tunnel selection process. The destination lockout period is a global value and is not individually configurable for particular destinations, tunnels, or tunnel groups.

NOTE: In general, a locked destination cannot be used until the lockout timer expires. However, when L2TP performs the tunnel selection process, in some circumstances it clears the lockout timer for a locked destination. See *Selection When Failover Between Preference Levels Is*

Configured and Selection When Failover Within a Preference Level Is Configured in LAC Tunnel Selection Overview for detailed information about the selection process.

BEST PRACTICE: Configure the lockout timeout to be equal to or shorter than the destruct timeout. Otherwise, the destruct timeout expires before the lockout timeout. In this event, the locked-out destination is destroyed and can be subsequently returned to service before the lockout timeout expires, thus negating the effectiveness of the lockout timeout.

To configure the destination lockout timeout:

- Specify the period in seconds.

```
[edit services l2tp destination]
user@host# set lockout-timeout seconds
```

The `show services l2tp destination lockout` command displays the destination lockout list and for each destination indicates how much time remains before its timeout expires. The `show services l2tp destination detail` command indicates for each destination whether it is locked and waiting for the timeout to expire or not locked.

Removing an L2TP Destination from the Destination Lockout List

When a PPP subscriber tries to log in to a domain, L2TP selects a tunnel associated with a destination in that domain and attempts to access the destination. If the connection attempt fails, L2TP places the destination on the destination lockout list. Destinations on this list are excluded from being considered for subsequent connections for a configurable period called the *destination lockout timeout*.

You can issue the `request services l2tp destination unlock` command for a particular destination to remove it from the destination lockout list. The result is that this destination is immediately available for consideration when a subscriber logs in to the associated domain.

To remove a destination from the destination lockout list:

- Specify the name of the destination to be unlocked.

```
user@host> request services l2tp destination unlock destination-name
```

Configuring L2TP Drain

For administrative purposes, you can set the state of an L2TP destination or tunnel to drain. This prevents the creation of new sessions, tunnels, and destinations at L2TP LAC and LNS.

You can configure L2TP drain at the global level or for a specific destination or tunnel. If the feature is configured at global L2TP level, then no new destination, tunnel, or session can be created. If the feature is configured for a specific destination, no new tunnel or session can be created at that destination. Similarly, if the feature is configured for a specific tunnel, no new sessions can be assigned to that tunnel, but new destinations and tunnels can be created.

- To prevent creation of new sessions, destinations, and tunnels for L2TP:

```
[edit services]
user@host# set l2tp drain
```

- To prevent creation of new tunnels and sessions at a particular destination:

```
[edit services]
user@host# set l2tp destination address ip-address drain
user@host# set l2tp destination address ip-address routing-instance routing-instance-name
drain
user@host# set l2tp destination name name drain
```

- To prevent creation of new sessions at a specific tunnel:

```
[edit services]
user@host# set l2tp tunnel name name drain
user@host# set l2tp tunnel name name address ip-address drain
user@host# set l2tp tunnel name name address ip-address routing-instance routing-instance-name
drain
```

NOTE: The tunnel *name* is the locally assigned name of the tunnel in the following format:

destination-name| tunnel-name or *tunnel-name*

When only the *tunnel-name* is provided, then you must include the address *ip-address* statement to identify the destination for the tunnel by.

When this feature is configured, the command output of `show services l2tp summary`, `show services l2tp destination`, and `show services l2tp tunnel` displays the state of the L2TP session, destination, and tunnel as Drain.

Using the Same L2TP Tunnel for Injection and Duplication of IP Packets

You can configure the same L2TP tunnel that is used for subscriber secure policy mirroring to be used for duplication of packets. Packets duplicated are used to inject traffic towards the customer or towards the network. Injection or transmission of packets is supported for all subscriber access modes. A single L2TP tunnel is used for both transmission of packets and duplication of packets. A port or interface that is configured for duplication of packets on one side of an L2TP tunnel is connected to the other tunnel endpoint. The other endpoint of the tunnel can send IP packets using the L2TP tunnel to the port or interface configured for packet duplication, and the IP packets received at that interface can be either forwarded to the customer or sent as though it has been received from the customer.

The remote tunnel endpoint sends an IP tunnel packet that contains an Ethernet MAC address in the payload. If the destination MAC address of the payload packet contains the MAC address of the router, the Ethernet packet is sent in the outgoing direction towards the network, and it is processed and forwarded as though it is received on the customer port. If the source MAC address of the payload packet contains the MAC address of the router, the Ethernet packet is transmitted in the outgoing direction towards the customer port. If the tunnel does not contain the `receive-cookie` configured, packet injection does not happen. In such a case, any received tunnel packet is counted and dropped in the same manner in which packets that arrive with a wrong cookie are counted and dropped.

To configure the packet to be duplicated and sent towards the customer or the network (based on the MAC address in the Ethernet payload), include the `decapsulate l2tp output-interface interface-name cookie l2tpv3-cookie` statement at the `[edit firewall family family-name filter filter-name term term-name then]` hierarchy level. You can also configure a counter for the duplicated or decapsulated L2TP packets by including the `count counter-name` statement at the `[edit firewall family family-name filter filter-name term term-name then]` hierarchy level

RELATED DOCUMENTATION

[L2TP for Subscriber Access Overview | 144](#)

[Configuring an L2TP LAC | 176](#)

[Configuring an L2TP LNS with Inline Service Interfaces | 259](#)

Specifying a Tunnel Switch Profile in a Domain Map

[LAC Tunnel Selection Overview | 183](#)

L2TP LAC Subscriber Configuration

IN THIS SECTION

- [Configuring an L2TP LAC | 176](#)
- [Configuring How the LAC Responds to Address and Port Changes Requested by the LNS | 177](#)
- [LAC Interoperation with Third-Party LNS Devices | 180](#)
- [Globally Configuring the LAC to Interoperate with Cisco LNS Devices | 181](#)

Configuring an L2TP LAC

To configure an L2TP LAC:

1. Configure a tunnel profile to apply to subscribers.
See ["Configuring a Tunnel Profile for Subscriber Access" on page 210.](#)
2. (Optional) Configure the method used for selecting among multiple tunnels.
 - See ["Configuring the L2TP LAC Tunnel Selection Parameters" on page 213.](#)
 - See ["Configuring Weighted Load Balancing for LAC Tunnel Sessions" on page 214.](#)
 - See ["Configuring Destination-Equal Load Balancing for LAC Tunnel Sessions" on page 215.](#)
 - See ["Configuring LAC Tunnel Selection Failover Within a Preference Level" on page 213.](#)
3. (Optional) Configure the LAC to not send Calling Number AVP 22 to the LNS.
See ["Preventing the LAC from Sending Calling Number AVP 22 to the LNS" on page 250.](#)
4. (Optional) Specify the method for setting the transmit and receive connect speeds.
See ["Configuring the Method to Derive the LAC Connection Speeds Sent to the LNS" on page 242.](#)
5. (Optional) Configure whether the L2TP failover protocol is negotiated or the silent failover method is used for resynchronization.
See ["Configuring the L2TP Peer Resynchronization Method" on page 324.](#)
6. (Optional) Specify the format for the tunnel name.
See ["Setting the Format for the Tunnel Name" on page 209.](#)
7. (Optional) Specify when and how many times L2TP retransmits unacknowledged control messages.
See ["Configuring Retransmission Attributes for L2TP Control Messages" on page 152.](#)

8. (Optional) Specify how long a tunnel can remain idle before being torn down.
See ["Setting the L2TP Tunnel Idle Timeout" on page 171.](#)
9. (Optional) Specify the L2TP receive window size for the L2TP tunnel. The receive window size specifies the number of packets a peer can send before waiting for an acknowledgment from the router.
See ["Setting the L2TP Receive Window Size" on page 171.](#)
10. (Optional) Specify how long the router retains information about terminated dynamic tunnels, sessions, and destinations.
See ["Setting the L2TP Destruct Timeout" on page 172.](#)
11. (Optional) Specify how the LAC handles IP address or UDP port change requests.
See ["Configuring How the LAC Responds to Address and Port Changes Requested by the LNS" on page 177.](#)
12. (Optional) Configure all tunnels on the LAC for interoperation with Cisco LNS devices.
See ["Globally Configuring the LAC to Interoperate with Cisco LNS Devices" on page 181.](#)
13. (Optional) Specify that the LAC sends information to the LNS about subscriber access lines.
See ["Configuring the Reporting and Processing of Subscriber Access Line Information" on page 245.](#)
14. (Optional) Configure the LAC to create the IPv6 address family (inet6) when establishing a tunnel for subscribers, enabling the application of IPv6 firewall filters.
See ["Enabling the LAC for IPv6 Services" on page 215.](#)
15. (Optional) Prevent the creation of new sessions, destinations, or tunnels for L2TP.
See ["Configuring L2TP Drain" on page 174.](#)
16. (Optional) Enable SNMP statistics counters.
See ["Enabling Tunnel and Global Counters for SNMP Statistics Collection" on page 154.](#)
17. (Optional) Configure trace options for troubleshooting the configuration.
See ["Tracing L2TP Events for Troubleshooting" on page 326.](#)

Configuring How the LAC Responds to Address and Port Changes Requested by the LNS

An LNS can use the SCCRPs message that it sends the LAC when a tunnel is being established to request a change in the destination IP address or UDP port that the LAC uses to communicate with the LNS. By default, the LAC accepts the request and makes the change. You can use the `tx-address-change` statement to configure one of the following methods for the LAC to handle these change requests for all tunnels:

- `accept`—The LAC accepts the change from the LNS. It sends all subsequent packets to and receives packets from the new IP address or UDP port.

- **ignore**—The LAC continues to send packets to the original address or port, but accepts packets from the new address or port.
- **reject**—The LAC sends a StopCCN message to the original address or port and then terminates the connection to that LNS.

The LAC accepts a change in address or port only once, when the tunnel is being established. Tunnels that are already established are not affected. The LAC drops any L2TP control packets containing change requests received at any other time, or in any packet other than an SCCRP message.

NOTE: This statement does not support IPv6 addresses.

To configure how the LAC handles change requests for the IP address, the UDP port, or both:

- (Optional) Configure the LAC to accept all change requests. This is the default behavior.

```
[edit services l2tp tunnel]
user@host# set tx-address-change accept
```

- (Optional) Configure the LAC to ignore all change requests.

```
[edit services l2tp tunnel]
user@host# set tx-address-change ignore
```

- (Optional) Configure the LAC to ignore change requests only for the IP address.

```
[edit services l2tp tunnel]
user@host# set tx-address-change ignore-ip-address
```

- (Optional) Configure the LAC to ignore change requests only for the UDP port.

```
[edit services l2tp tunnel]
user@host# set tx-address-change ignore-udp-port
```

- (Optional) Configure the LAC to reject all change requests.

```
[edit services l2tp tunnel]
user@host# set tx-address-change reject
```

- (Optional) Configure the LAC to reject change requests only for the IP address.

```
[edit services l2tp tunnel]
user@host# set tx-address-change reject-ip-address
```

- (Optional) Configure the LAC to reject change requests only for the UDP port.

```
[edit services l2tp tunnel]
user@host# set tx-address-change reject-udp-port
```

For example, the following configuration causes the LAC to ignore requests to change the UDP port, but to reject requests to change the IP address:

```
[edit services l2tp tunnel]
user@host# set tx-address-change ignore-udp-port
user@host# set tx-address-change reject-ip-address
```

NOTE: Conflicting configurations are not allowed and fail the configuration commit check. You cannot For example, the following configuration fails, because it specifies that UDP port changes are ignored, but that *all* changes are rejected:

```
[edit services l2tp tunnel]
user@host# set tx-address-change ignore-udp-port
user@host# set tx-address-change reject
```

Use the `show services l2tp summary` command to display the current behavior of the LAC:

```
show services l2tp summary
Failover within a preference level is Disabled
Weighted load balancing is Disabled
Tunnel authentication challenge is Enabled
```

```

Calling number avp is Enabled
Failover Protocol is Disabled
Tx Connect speed method is static
Rx speed avp when equal is Disabled
Tunnel assignment id format is assignment-id
Tunnel Tx Address Change is Ignore
Max Retransmissions for Established Tunnel is 7
Max Retransmissions for Not Established Tunnel is 5
Tunnel Idle Timeout is 60 seconds
Destruct Timeout is 300 seconds
Destination Lockout Timeout is 300 seconds
Destinations: 1, Tunnels: 0, Sessions: 0

```

Depending on the configuration, this command displays one of the following outputs:

```

Tunnel Tx Address Change is Accept
Tunnel Tx Address Change is Ignore
Tunnel Tx Address Change is Reject
Tunnel Tx Address Change is Ignore IP Address & Accept UDP Port
Tunnel Tx Address Change is Ignore IP Address & Reject UDP Port
Tunnel Tx Address Change is Accept IP Address & Ignore UDP Port
Tunnel Tx Address Change is Accept IP Address & Reject UDP Port
Tunnel Tx Address Change is Reject IP Address & Accept UDP Port
Tunnel Tx Address Change is Reject IP Address & Ignore UDP Port

```

LAC Interoperation with Third-Party LNS Devices

In some network environments, the LAC may need to interoperate with an LNS configured on a device from another vendor that does not run Junos OS. Interoperation with Cisco Systems devices requires the LAC to communicate a NAS port type, but the LAC does not provide this information by default.

You can enable interoperation with Cisco Systems devices by configuring the NAS port method as `cisco-avp`, which causes the LAC to include the Cisco Systems NAS Port Info AVP (100) when it sends an incoming call request (ICRQ) to the LNS. The AVP includes information that identifies the NAS port and indicates whether the port type is ATM or Ethernet.

You can configure the NAS port method globally for all tunnels on the LAC or in a tunnel profile for only the tunnels instantiated by the profile.

You can also include the Tunnel-Nas-Port-Method VSA [26–30] in your RADIUS server configuration with the value set to 1 to indicate Cisco Systems CLID. In this case, RADIUS can override the global value by modifying or creating a tunnel profile. The RADIUS configuration has precedence over the tunnel profile configuration, which in turn has precedence over the global LAC configuration.

If the LNS receiving the AVP is an MX Series router instead of a Cisco Systems device, the LNS simply ignores the AVP, unless the LNS is configured for L2TP tunnel switching. In that case, the LNS preserves the value of the AVP and passes it along when it switches tunnels for the LAC.

Globally Configuring the LAC to Interoperate with Cisco LNS Devices

Cisco LNS devices require from the LAC both the physical NAS port number identifier and the type of the physical port, such as Ethernet or ATM. By default, the LAC does not include this information. You can globally configure the LAC to provide this information by including the NAS Port Info AVP (100) in the ICRQ that it sends to the LNS. This configuration enables the LAC to interoperate with a Cisco LNS.

To globally configure the LAC to include the NAS Port Info AVP:

- Specify the NAS port method.

```
[edit services l2tp tunnel]
user@host# set nas-port-method cisco-avp
```

NOTE: This global configuration for the LAC can be overridden by the configuration in a tunnel profile or RADIUS.

Use the `show services l2tp tunnel extensive` command to display the current behavior of the LAC:

```
show services l2tp tunnel extensive
Tunnel local ID: 51872, Tunnel remote ID: 8660
  Remote IP: 192.0.2.20:1701
  Sessions: 5, State: Established
  Tunnel Name: 1/tunnel-test-2
  Local IP: 203.0.113.2:1701
  Local name: testlac, Remote name: ce-lns
  Effective Peer Resync Mechanism: silent failover
Nas Port Method: none
  Tunnel Logical System: default, Tunnel Routing Instance: default
```

Max sessions: 128100, Window size: 4, Hello interval: 60
 Create time: Thu Jul 25 12:55:41 2013, Up time: 11:18:14
 Idle time: 00:00:00

Statistics since: Thu Jul 25 12:55:41 2013

	Packets	Bytes
Control Tx	702	15.5k
Control Rx	690	8.5k
Data Tx	153.3k	6.6M
Data Rx	126.3k	5.9M
Errors Tx	0	
Errors Rx	0	

RELATED DOCUMENTATION

[L2TP for Subscriber Access Overview | 144](#)

[Configuring an L2TP LAC | 176](#)

[Configuring a Tunnel Profile for Subscriber Access | 210](#)

Juniper Networks VSAs Supported by the AAA Service Framework

L2TP LAC Tunneling for Subscribers

IN THIS SECTION

- [LAC Tunnel Selection Overview | 183](#)
- [L2TP Session Limits Overview | 200](#)
- [Limiting the Number of L2TP Sessions Allowed by the LAC or LNS | 206](#)
- [Setting the Format for the Tunnel Name | 209](#)
- [Configuring a Tunnel Profile for Subscriber Access | 210](#)
- [Configuring the L2TP LAC Tunnel Selection Parameters | 213](#)
- [Configuring LAC Tunnel Selection Failover Within a Preference Level | 213](#)
- [Configuring Weighted Load Balancing for LAC Tunnel Sessions | 214](#)
- [Configuring Destination-Equal Load Balancing for LAC Tunnel Sessions | 215](#)
- [Enabling the LAC for IPv6 Services | 215](#)

- [Testing L2TP Tunnel Configurations from the LAC | 216](#)

LAC Tunnel Selection Overview

IN THIS SECTION

- [Selection When Failover Between Preference Levels Is Configured | 186](#)
- [Selection When Failover Within a Preference Level Is Configured | 191](#)
- [Selection When Distributing the Session Load Across Multiple LNSs | 193](#)

When a user logs in to a domain, the PPP client contacts the LAC to establish a connection. The LAC has to find a destination in the domain and a tunnel that can reach it. The association between destinations, tunnels, and domains is provided by a tunnel profile either in a domain map in the subscriber's access profile or in the Tunnel-Group attribute (VSA 26-64) received from a RADIUS server. The RADIUS attribute takes precedence over a profile specified in a domain map. The tunnel profile includes a list of tunnels; each tunnel is associated with a destination IP address and with a tunnel preference level.

L2TP enables you to specify:

- Up to 31 destinations for a domain.
- Up to eight levels of tunnel preference. The preference level determines the order in which the LAC attempts to use an existing tunnel (or establish a new one) to a destination in the user's requested domain.

NOTE: Zero (0) is the highest level of preference; this is the most-preferred level.

If two tunnels both reach valid destinations within a domain, the LAC first selects the tunnel with the highest preference level. For example, when Tunnel A has a preference level of 1 and Tunnel B has a preference level of 4, the LAC attempts to use Tunnel A first.

- Up to 31 destinations for a single preference level.

When the LAC determines that a PPP session should be tunneled, it selects a tunnel from the set of tunnels associated with either the PPP user or the PPP user's domain by a tunnel profile.

Tunnel selection is affected by the following configurations:

- **Failover between preference levels**—By default, when a tunnel to a valid destination is not selected within a preference level, the selection process fails over to the next level; that is, the LAC drops down to the next lower level to continue the search for a suitable tunnel. See ["Selection When Failover Between Preference Levels Is Configured"](#) on page 186 for more information.
- **Failover within a preference level**—In this case, the LAC does not limit its attempts to establish a session to only a single tunnel at a preference level. If the attempt fails through the selected tunnel, the selection process fails over within that same level by selecting another suitable tunnel to a valid destination. The LAC continues its connection attempts within the level until no more tunnels to a valid destination are available at that level. Then the LAC drops down to the next lower level to continue the search. See ["Selection When Failover Within a Preference Level Is Configured"](#) on page 191 for more information.
- **Maximum sessions per tunnel**—When the maximum number of sessions allowed per tunnel is configured, the LAC takes that setting into account during the tunnel selection process. The maximum number of sessions per tunnel can be configured by means of the RADIUS Tunnel-Max-Sessions VSA [26-33] or by including the `max-sessions` statement in a tunnel profile.

When a randomly selected tunnel has a current session count equal to its maximum session count, the LAC does not attempt to connect to a destination with that tunnel. Instead, it selects an alternate tunnel from the set of tunnels at that preference level that have valid destinations in the domain. If no such tunnels exist at the current preference level, the LAC drops to the next preference level to make the selection. This process is consistent, regardless of which failover scheme is currently running on the LAC.

When the maximum number of sessions is not configured for a tunnel, then that tunnel has no upper limit on the number of sessions it can support. By default, the maximum sessions value is 0 (zero), which allows unlimited sessions in the tunnel.

- **Weighted load balancing**—This balancing method uses a probability-based evaluation of tunnel weight to distribute sessions across tunnels. The LAC still selects tunnels randomly within a preference level, but on average the sessions are distributed across tunnels in relationship to the weight of the tunnels. The weight of a tunnel is determined by the tunnel's maximum session limit and the maximum session limits of the other tunnels at the same preference level. See ["Weighted Load Balancing"](#) on page 194 for more information.
- **Destination-equal load balancing**—This session-balancing method evaluates tunnels according to the number of sessions to the destination and the number of sessions carried by the tunnel in order to spread the session load equally among all tunnels. The tunnel with a destination that has the lowest session count is determined to have the lightest load. This process operates on tunnels at the highest available preference level. See ["Destination-Equal Load Balancing"](#) on page 195 for more information.

Take the following information into consideration to understand the tunnel and destination selection process and failover:

- More than one tunnel may be able to reach a destination, and those tunnels can have the same preference level or different preference levels.
- The tunnel selected to establish the subscriber session may itself already be established, meaning that it has currently active sessions. Alternatively, the LAC might have to establish a new tunnel to the destination if no tunnel capable of reaching the destination is already established.
- A *valid* destination meets the following criteria:
 - It is reachable by a tunnel that has not met its maximum session limit.
 - It has not yet been contacted for the current subscriber login request.
 - It can be either locked or unlocked.
- A *locked* destination is one for which the destination lockout timer is running. Locked destinations are placed on a lockout list until the timer expires or is cleared (reset to zero). Destinations on the list cannot be contacted to establish a session.
- An *unlocked* destination is one for which the destination lockout timer is zero.
- When the LAC discovers valid destinations that are locked, it places them on the DestinationsLockedNotContacted list, which is different than the lockout list that includes all locked-out destinations. The DestinationsLockedNotContacted list includes only locked destinations that the LAC has not yet attempted to contact for the current, in-progress subscriber login. The DestinationsLockedNotContacted list does not include destinations that the LAC locks out after it has attempted and failed to establish a connection.
- You can use the `clear services l2tp destination lockout` command to manually clear all locked destinations or only locked destinations that match the specified local or remote gateway address. You might use the `if` command if, for example, you want to clear a specific destination so that it gets priority within a preference level.
- The failover behavior that is part of the tunnel selection process applies only when the destination is unreachable for one of the following reasons:
 - The LNS fails to return an SCCRP message in response to the SCCRQ message from the LAC after the maximum number of retransmission attempts.
 - The tunnel is established, but the LNS does not return an ICRP message in response to the ICRQ from the LAC after the maximum number of retransmission attempts.
- This failover behavior does not apply in the following circumstances:
 - The client terminates the connection.
 - The tunnel is established, but the LNS sends a CDN message while the LAC is attempting to establish the session with the LNS, resulting in the failure of the subscriber login attempt.

Selection When Failover Between Preference Levels Is Configured

When a user tries to log in to a domain in a default configuration—that is, when failover within a preference level and load balancing are not configured—the LAC searches for valid destinations to the requested domain, starting at the highest tunnel preference level. If no valid destination is found, or the attempt to connect to a destination fails, the LAC drops down to the next lower level to continue searching. The search process is the same for all levels except for the lowest:

1. The search begins by identifying tunnels with valid destinations at the preference level from among all the tunnels specified in the domain's tunnel profile.
2. All locked, valid destinations are placed on the DestinationsLockedNotContacted list. No attempt is made to contact any of these destinations.
3. From among the unlocked, valid destinations, the LAC selects one at random and attempts to connect through the associated tunnel; if the tunnel has no current sessions, then the LAC must establish the tunnel.

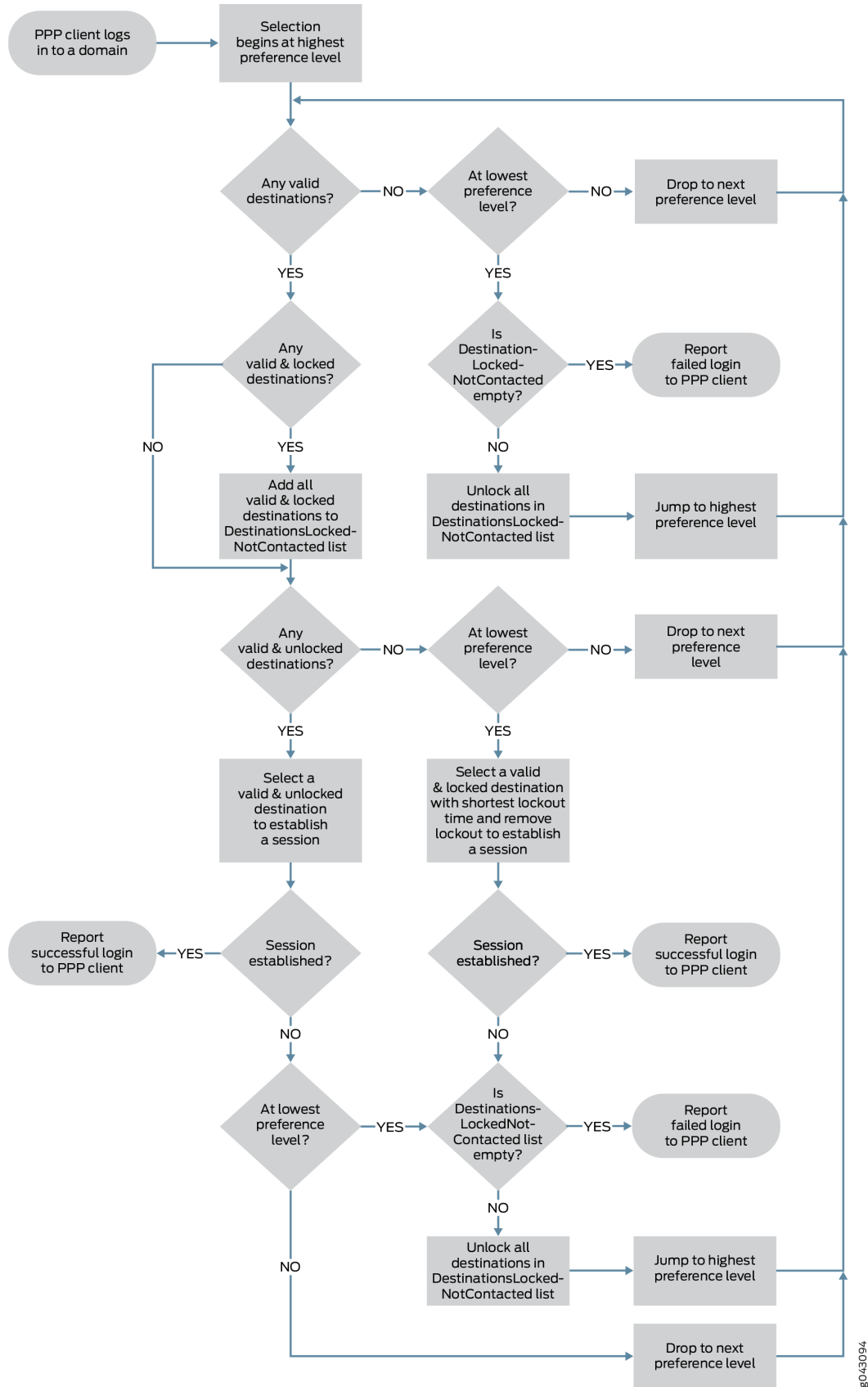
NOTE: Random selection is the default behavior. The behavior is different when weighted load balancing or destination-equal load balancing is configured. See "[Selection When Distributing the Session Load Across Multiple LNSs](#)" on page 193 for information about load balancing.

- If the attempt is successful, the LAC reports the successful login to the PPP client. The LAC also clears all destinations on the DestinationsLockedNotContacted list.
 - If the LAC receives no response, it retries the attempt up to the maximum retry number. If the LAC exhausts the retries without receiving a reply, the attempt is considered unsuccessful and the LAC marks the destination as unreachable by locking out the destination. It places the destination on the lockout list and starts the destination lockout timer.
4. What the LAC does next depends on the current preference level.
 - If it is not the lowest preference level, then the LAC drops to the next lower preference level and continues the search process.
 - If it is the lowest preference level and the DestinationsLockedNotContacted list is not empty, then the LAC unlocks all destinations in the DestinationsLockedNotContacted list and jumps back up to the highest preference level and restarts the search process.
 - If it is the lowest preference level and the DestinationsLockedNotContacted list is empty—meaning that all valid destinations have been attempted—then the LAC reports a failed login to the PPP client.

5. When the valid destinations at one level are all locked, what the LAC does next depends on the current preference level.
 - If it is not the lowest preference level, then the LAC drops to the next lower preference level and continues the search process.
 - If it is the lowest preference level, the LAC selects the locked, valid destination with the shortest remaining lockout time. It clears the lockout timer and attempts to connect to the destination and establish a session.
 - If the attempt is successful, the LAC reports the successful login to the PPP client.
 - If the attempt fails and the DestinationsLockedNotContacted list is empty—meaning that all valid destinations have been attempted—then the LAC reports a failed login to the PPP client.
 - If the attempt fails and the DestinationsLockedNotContacted list is not empty, then the LAC unlocks all destinations in the DestinationsLockedNotContacted list, jumps back up to the highest preference level, and restarts the search process.
6. When no valid destinations are present, what the LAC does next depends on the current preference level.
 - If it is not the lowest preference level, then the LAC drops to the next lower preference level and continues the search process.
 - If it is the lowest preference level and the DestinationsLockedNotContacted list is empty—meaning that all valid destinations have been attempted—then the LAC reports a failed login to the PPP client.
 - If it is the lowest preference level and the DestinationsLockedNotContacted list is not empty, then the LAC unlocks all destinations in the DestinationsLockedNotContacted list, jumps back up to the highest preference level, and restarts the process.
7. The search and failover process cycles through the levels until either a session is established or all valid destinations have been attempted—no destinations remain on the DestinationsLockedNotContacted list—and the login fails.

[Figure 15 on page 188](#) illustrates the possible conditions and decision points that determine the selection of a destination and corresponding tunnel for the default case, where failover occurs between tunnel preference levels.

Figure 15: Destination and Tunnel Selection Process with Failover Between Preference Levels



For example, suppose that the tunnel profile includes the following tunnels, each with a valid destination:

- Preference 0, Tunnel 1, 192.168.10.10
- Preference 1, Tunnel 2, 192.168.22.22
- Preference 1, Tunnel 3, 192.168.33.33
- Preference 2, Tunnel 4, 192.168.44.44

Failover within preference and load balancing are not configured.

When a PPP user tries to connect to the domain, the LAC acts as follows:

1. At the highest preference level, 0, the LAC selects Tunnel 1 because it is the only tunnel in the level with a valid destination. The LAC attempts to reach 192.168.10.10.
2. This connection attempt fails, so the LAC locks out 192.168.10.10. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
3. The LAC drops (fails over) to the next level, preference level 1, to reach a destination for the domain. The LAC randomly selects between 192.168.22.22 through Tunnel 2 and 192.168.33.33 through Tunnel 3. It selects 192.168.22.22 and attempts to connect through Tunnel 2.
4. The connection attempt to 192.168.22.22 fails, so the LAC locks out 192.168.22.22. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.

NOTE: Even though Tunnel 3 has an unlocked, valid destination, the LAC cannot now select that tunnel to reach 192.168.33.33, because the LAC can make only one attempt to reach a valid destination each time it searches in a level when the failover method is between preference levels.

5. The LAC drops to the final (lowest) level in this example, preference level 2. The LAC selects Tunnel 4 because it is the only tunnel in the level with a valid destination. The LAC attempts to reach 192.168.44.44.
6. The connection attempt to 192.168.44.44 also fails, so the LAC locks out 192.168.44.44. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
7. Because this is the lowest level, and the DestinationsLockedNotContacted list is empty, the LAC rejects the login request from the PPP client.

Destinations 192.168.10.10, 192.168.22.22, and 192.168.44.44 were locked out, but not added to the DestinationsLockedNotContacted list because the LAC locked them out after attempting to connect. Destination 192.168.33.33 was not contacted, but not added to the DestinationsLockedNotContacted list because it is not locked out.

8. The client tries to log in again and the LAC repeats the tunnel selection process, starting over at preference level 0 to check for an unlocked, valid destination, and cycling through the levels as needed.
9. At preference level 0, 192.168.10.10 is the only valid destination and is still locked out, so the LAC cannot attempt to connect to the destination. The LAC adds 192.168.10.10 to the DestinationsLockedNotContacted list and then drops to preference level 1.

NOTE: Remember that the destination lockout timer applies globally, so it persists across multiple subscriber logins. The DestinationsLockedNotContacted list applies only to a given subscriber login and does not persist. Even though the LAC contacted 192.168.10.10 for this subscriber, it was during a previous login attempt. In this login attempt, it cannot contact the destination because of the lockout, and consequently places the destination on the DestinationsLockedNotContacted list.

10. At preference level 1, 192.168.22.22 is still locked out, so the LAC adds 192.168.22.22 to the DestinationsLockedNotContacted list. 192.168.33.33 is still available. The LAC attempts to connect to 192.168.33.33 through Tunnel 3.
11. This connection attempt fails, so the LAC locks out 192.168.33.33. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires. The LAC drops to preference level 2.
12. 192.168.44.44 is still locked out, so the LAC adds 192.168.44.44 to the DestinationsLockedNotContacted list.
13. This is the lowest preference level, but this time the DestinationsLockedNotContacted list is not empty; it contains 192.168.10.10, 192.168.22.22, and 192.168.44.44. The LAC unlocks all destinations on the DestinationsLockedNotContacted list and then jumps back to the highest preference level.
14. At preference level 0, the LAC attempts to connect to 192.168.10.10 because it was unlocked. The LAC establishes the session and reports the successful login to the PPP client.

Although the LAC does not attempt to contact a destination that is locked out, there is a special case when the LAC has reached the lowest preference level. The level must have more than one valid destination and all of them must be locked out. For example, suppose that the tunnel profile includes the following tunnels, each with a valid destination:

- Preference 0, Tunnel 1, 192.168.10.10
- Preference 1, Tunnel 2, 192.168.22.22. The destination is locked out with the lockout timer currently at 245 seconds.
- Preference 1, Tunnel 3, 192.168.33.33. The destination is locked out with the lockout timer currently at 180 seconds.

Failover within preference and load balancing are not configured.

When a PPP user tries to connect to the domain, the LAC acts as follows:

1. At the highest preference level, 0, the LAC selects Tunnel 1 because it is the only tunnel in the level with a valid destination. The LAC attempts to reach 192.168.10.10.
2. This connection attempt fails, so the LAC locks out 192.168.10.10. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
3. The LAC drops to the next level, preference level 1, to reach a destination for the domain. Both valid destinations at this level, 192.168.22.22 and 192.168.33.33, are locked out.
4. The LAC adds both destinations to the DestinationsLockedNotContacted list.
5. Because this is the lowest preference level, the LAC determines which destination has a shorter remaining lockout time. It selects 192.168.33.33 because it has a shorter remaining lockout time (180 seconds) than 192.168.22.22 (245 seconds). The LAC unlocks 192.168.33.33 and attempts to connect through Tunnel 3. As a consequence, the LAC also removes 192.168.33.33 from the DestinationsLockedNotContacted list.
6. The connection attempt is successful and a session is established to 192.168.33.33. The LAC reports a successful login to the PPP client.

Selection When Failover Within a Preference Level Is Configured

When you configure failover *within* a preference level, the destination and tunnel selection process is the same as for the default configuration, with one exception: the LAC is not limited to only one connection attempt at a preference level.

When the LAC tries to connect to an unlocked, valid destination and is unsuccessful, it locks out that destination but does not immediately drop down to the next lower level. Instead, if another unlocked, valid destination is available at the same preference level, the LAC attempts to connect to that destination.

If the LAC does not connect, then it continues to try to reach a destination within that preference level until no more unlocked, valid destinations remain to be attempted. At that point the LAC drops down to

search at the next lower preference level. At each level, the LAC searches for and attempts to connect to a valid destination until no unlocked, valid destinations are available.

If the LAC drops down to the lowest preference level and finds no unlocked, valid destinations, the behavior depends on the DestinationsLockedNotContacted list:

- If the DestinationsLockedNotContacted list is not empty, then the LAC unlocks all destinations in the DestinationsLockedNotContacted list and jumps back up to the highest preference level and restarts the search process.
- If the DestinationsLockedNotContacted is empty—meaning that all valid destinations have been attempted—then the LAC reports a failed login to the PPP client.

For example, suppose that the tunnel profile specifies the following tunnels and destinations. Load balancing is not configured. All destinations are valid; all are unlocked except 192.168.3.3. The preference levels for the tunnels are assigned as follows:

- Preference 0, Tunnel 1, 192.168.1.1, unlocked
- Preference 0, Tunnel 2, 192.168.2.2, unlocked
- Preference 0, Tunnel 3, 192.168.3.3, lockout timer 100 seconds
- Preference 1, Tunnel 4, 192.168.4.4, unlocked
- Preference 1, Tunnel 5, 192.168.5.5, unlocked

In this example, when a PPP user tries to connect to the domain, the LAC acts as follows:

1. The LAC randomly selects between the two unlocked, valid destinations at preference level 0, 192.168.1.1 through Tunnel 1 and 192.168.2.2 through Tunnel 2. It chooses 192.168.2.2 and attempts to connect through Tunnel 2.
2. The connection attempt to 192.168.2.2 fails, so the LAC locks out 192.168.2.2. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
3. The LAC then attempts to connect to 192.168.1.1 through Tunnel 1 at preference level 0.
4. The connection attempt to 192.168.1.1 fails, so the LAC locks out 192.168.1.1. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
5. 192.168.3.3 through Tunnel 3 is the only remaining valid destination at preference level 0, but it is locked. The LAC adds 192.168.3.3 to the DestinationsLockedNotContacted list. The LAC did not add 192.168.1.1 and 192.168.2.2 to the DestinationsLockedNotContacted list, because it locked them out after attempting to contact them.

6. Because level 0 has no more unlocked, valid destinations, the LAC drops to the next level, preference level 1, to reach a destination for the domain.
7. At preference level 1, the LAC randomly selects 192.168.4.4 and attempts to connect through Tunnel 4.
8. The connection attempt to 192.168.4.4 fails, so the LAC locks out 192.168.4.4. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
9. The LAC then attempts to connect to 192.168.5.5 through Tunnel 5 at preference level 1.
10. The connection attempt to 192.168.5.5 fails, so the LAC locks out 192.168.5.5. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires. Level 1 has no more unlocked, valid destinations. Because the DestinationsLockedNotContacted list is not empty, the LAC unlocks all the destinations on the list—in this case, 192.168.3.3—and jumps back up to the highest preference level, 0.
11. 192.168.3.3 is now the only unlocked destination at preference level 0, so the LAC attempts to connect to it through Tunnel 3.
12. The connection attempt to 192.168.3.3 fails, so the LAC locks out 192.168.3.3. It is not considered again during this login attempt, and cannot be considered for any login attempt until the destination lockout timer expires.
13. Because level 0 has no more unlocked, valid destinations, the LAC drops to the next level, preference level 1.
14. Preference level 1 has no unlocked, valid destinations. The DestinationsLockedNotContacted is empty because the LAC has contacted all valid destinations at both preference levels. The LAC rejects the login request from the PPP client.

Selection When Distributing the Session Load Across Multiple LNSs

Multiple tunnel profiles can be configured on the LAC; some tunnels may share destinations. When the LAC tunnels the session for a PPP subscriber to the LNS, a tunnel has to be selected for the subscriber session. The tunnel selection process chooses a tunnel with the highest preference that has a reachable destination. By default, the LAC selects a tunnel at random from among multiple tunnels that meet the same criteria. Alternatively, you can configure load balancing to enable different selection choices. Both load-balancing methods affect which tunnels and destinations the LAC selects, but the selection and failover process otherwise remains the same.

NOTE: Weighted load balancing and destination-equal load balancing are mutually exclusive. You can enable only one or the other.

Weighted Load Balancing

Weighted load balancing evaluates tunnels according to their weight. The weight of a tunnel is determined by the tunnel's maximum session limit and the maximum session limits of the other tunnels at the same preference level. The tunnel with the highest maximum session limit has the highest weight in that preference level. The tunnel with the next-highest maximum session limit has the next-highest weight, and so on. The tunnel with the lowest maximum session limit has the lowest weight.

NOTE: Tunnel selection and session distribution are probability based; the load is not strictly distributed according to weight.

When you configure weighted load balancing, the LAC still selects tunnels randomly within a preference level, but on average the sessions are distributed across tunnels in relationship to the weight of the tunnels.

With weighted load balancing, the LAC generates a random number within a range equal to the aggregate total of all session limits for all tunnels in the preference level. It associates part of the range—a pool of numbers—with each tunnel proportional to the tunnel weight. A tunnel with a higher weight is associated with a greater portion of the range—a larger pool—than a tunnel with a lower weight. A tunnel is selected when the random number is in its associated pool of numbers. The random number is more likely, on average, to be in a larger pool, so a tunnel with a higher weight (larger pool) is more likely to be selected than a tunnel with a lower weight (smaller pool).

For example, consider a preference level that has only two tunnels, 1 and 2. Tunnel 1 has a maximum limit of 1000 sessions and Tunnel 2 has a limit of 2000 sessions, resulting in an aggregate total of 3000 sessions. The LAC generates a random number from a pool of 3000 in the range from 0 through 2999. A pool of 1000 numbers, the portion of the range from 0 through 999, is associated with Tunnel 1. A pool of 2000 numbers, the portion of the range from 1000 through 2999, is associated with Tunnel 2.

- When the generated number is less than 1000, then Tunnel 1 is selected, even though it has a lower weight (1000) than Tunnel 2 (2000).
- When the generated number is 1000 or larger, then Tunnel 2 is selected.

Because the pool of possible generated numbers for Tunnel 2 (2000) is twice that for Tunnel 1 (1000), Tunnel 2, *on average*, is selected twice as often as Tunnel 1.

Destination-Equal Load Balancing

Destination-equal load balancing evaluates tunnels according to the number of sessions to the destination and the number of sessions carried by the tunnel in order to spread the session load equally among all tunnels. The tunnel with a destination that has the lowest session count is considered to have the lightest load. This process operates on tunnels at the highest available preference level and uses the following guidelines:

- When each tunnel goes to a separate destination and only one destination has the lowest session count among all destinations, the LAC selects the tunnel to that destination.
- When each tunnel goes to a separate destination and more than one destination has the same lowest session count, the LAC selects a tunnel at random from among the tunnels to these destinations.
- When more than one tunnel goes to the same destination and that destination has the lowest destination session count, the LAC selects from among these tunnels the one that has the lowest total number of tunnel sessions. If the tunnel session count is the same for all these tunnels, then the LAC selects one of them at random.

Consider the following scenarios to better understand tunnel selection behavior when destination-equal load balancing is enabled.

In Scenario 1, every tunnel has a different valid destination and only the destination session count is evaluated:

- Tunnel 1, preference level 1, 192.168.1.1, destination session count = 200
- Tunnel 2, preference level 1, 192.168.2.2, destination session count = 50
- Tunnel 3, preference level 1, 192.168.3.3, destination session count = 300
- Tunnel 4, preference level 1, 192.168.4.4, destination session count = 100

When the first PPP user tries to connect to the domain, the LAC selects Tunnel 2, because it is at the highest preference level, 1, and has the valid destination, B, with the lowest session count, 50.

When additional PPP users try to connect to the domain, the LAC acts as follows:

1. Tunnel 2 continues to be selected until the session count for 192.168.2.2 equals 100, matching the next lowest session count, 192.168.4.4's in Tunnel 4.
2. When the next subscriber logs in, the LAC randomly selects between Tunnel 2 and Tunnel 4, because their destinations have the same session count, and it is lower than that for the other destinations.
3. Whichever tunnel is selected from this pair, the session count for its destination is now 101. The other tunnel is selected when the next subscriber logs in, because it has the lower destination session count of 100. This raises its destination session count to 101, matching the other tunnel.

4. As subscribers continue to log in, the LAC repeats this process, randomly selecting between Tunnel 2 and Tunnel 4 when their session counts match and then selecting the other tunnel with the next subscriber, until their destination session counts both reach 200, matching Tunnel 1.
5. When the next subscriber logs in, the LAC now randomly selects among Tunnel 1, Tunnel 2, and Tunnel 4, because 192.168.1.1, 192.168.2.2, 192.168.3.3 all have the same session count of 200. The destination session count is raised for the selected tunnel to 201, so for the next subscriber, the LAC randomly selects between the other two tunnels. Now two tunnels have a destination session count of 201, so the LAC selects the remaining tunnel for the next subscriber.
6. As subscribers continue to log in, the LAC repeats this process, randomly selecting among Tunnel 1, Tunnel 2, and Tunnel 4 when their session counts match, randomly selecting between the remaining pair for the next subscriber, and then selecting the remaining tunnel, so the destination session counts for these three tunnels match again. This pattern continues until the destination session count for all three tunnels reaches 300, matching Tunnel 3.
7. Now the destinations for all four tunnels have the same session count. Because there are only four tunnels, the final pattern is established. The LAC first randomly selects among all four tunnels, then the remaining three, then the remaining pair, and finally selects the last tunnel. When the destination session counts are all the same, the LAC starts this pattern again.

In Scenario 2, two tunnels share the same valid destination. The tunnel session count and the destination session count are both evaluated:

- Tunnel 1, preference level 1, tunnel session count = 120, 192.168.1.1, destination session count = 200
- Tunnel 2, preference level 1, tunnel session count = 80, 192.168.1.1, destination session count = 200
- Tunnel 3, preference level 1, 192.168.2.2, destination session count = 300
- Tunnel 4, preference level 2, 192.168.3.3, destination session count = 100

When the first PPP user tries to connect to the domain, the LAC first selects between destinations. The tunnels for both 192.168.1.1 and 192.168.2.2 are at preference level 1. The LAC selects 192.168.1.1, because it has a lower session count (200) than 192.168.2.2 (300). The LAC then has to choose between Tunnel 1 and Tunnel 2 because both go to 192.168.1.1. The LAC evaluates the tunnel session count. Tunnel 2 has a lower count (80) than Tunnel 1 (120), so the LAC selects Tunnel 2 for the first subscriber.

When additional PPP users try to connect to the domain, the LAC acts as follows:

1. Tunnel 2 continues to be selected until its tunnel session count increases to 120, matching Tunnel 1.
2. When the next subscriber logs in, the LAC randomly selects between Tunnel 1 and Tunnel 2, because they have the same tunnel session count. The tunnel session count of the selected tunnel is raised to 121.

3. When the next subscriber logs in, the LAC selects the other tunnel to 192.168.1.1, because it has a lower tunnel session count. From this point, the LAC continues to alternate, first making a random selection between Tunnels 1 and 2 and then selecting the other tunnel, until the destination session count rises to 300, matching the session count for 192.168.2.2 in Tunnel 3. (At this point, the tunnel session count is 150 for both Tunnel 1 and Tunnel 2.)
4. For the next subscriber, the LAC randomly selects among Tunnels 1, 2, and 3.
 - If the LAC selects either Tunnel 1 or Tunnel 2, the 192.168.1.1 session count rises to 301. Consequently the LAC selects Tunnel 3 for the next subscriber because the 192.168.2.2 session count is still 300. At this point, both destinations have the same session count again.
 - If the LAC selects Tunnel 3, the 192.168.2.2 session count rises to 301. For the next subscriber, the LAC randomly selects between Tunnel 1 and Tunnel 2 because they both go to 192.168.1.1. Whichever one the LAC selects, the 192.168.1.1 session count rises to 301. At this point, both destinations have the same session count again.

NOTE: The tunnel session count for Tunnels 1 and 2 is no longer evaluated; the LAC only considers the destination session count for 192.168.1.1 and 192.168.2.2.

This pattern continues for all subsequent subscribers.

In Scenario 3, each tunnel has a different valid destination and only the destination session count is evaluated:

- Tunnel 1, preference level 1, 192.168.1.1, destination session count = 100
- Tunnel 2, preference level 1, 192.168.2.2, destination session count = 100
- Tunnel 3, preference level 1, 192.168.3.3, destination session count = 100
- Tunnel 4, preference level 1, 192.168.4.4, destination session count = 100

When the first PPP user tries to connect to the domain, the LAC determines that the destination session count is the same for all destinations for all four tunnels at the preference level. Consequently, the LAC selects randomly among the four tunnels.

Suppose the LAC selects Tunnel 1 for the first subscriber.

When additional PPP users try to connect to the domain, the LAC acts as follows:

1. The LAC selects randomly among Tunnels 2, 3, and 4, because Destinations 192.168.2.2, 192.168.3.3, and 192.168.4.4 all have the same session count, 100, which is lower than the current session count for 192.168.1.1, 101.

2. Suppose the LAC selects Tunnel 2. For the next subscriber, the LAC randomly selects between Tunnels 3 and 4, because 192.168.3.3 and 192.168.4.4 all have the same session count, 100, which is lower than the current session count of 101 for 192.168.1.1 and 192.168.2.2.
3. Suppose the LAC selects Tunnel 3. For the next subscriber, the LAC selects Tunnel 4, because 192.168.4.4 has a session count of 100, and all the other destinations have a count of 101.
4. Now the destinations for all four tunnels have the same session count. Because there are only four tunnels, the final pattern is established. As subscribers continue to log in, the LAC first randomly selects among all four tunnels, then the remaining three, then the remaining pair, and finally selects the last tunnel. When the destination session counts are all the same, the LAC starts this pattern again.

In Scenario 4, the LAC evaluates both destination session limits and tunnel maximum session limits:

- Tunnel 1, preference level 1, 192.168.1.1, destination session count = 30, tunnel maximum session limit = 200
- Tunnel 2, preference level 1, 192.168.2.2, destination session count = 40, tunnel maximum session limit = 200
- Tunnel 3, preference level 1, 192.168.3.3, destination session count = 300, tunnel maximum session limit = 1000
- Tunnel 4, preference level 2, 192.168.4.4, destination session count = 100

When the first PPP user tries to connect to the domain, the LAC selects Tunnel 1, because 192.168.1.1 has the lowest session count in the preference level.

When additional PPP users try to connect to the domain, the LAC acts as follows:

1. The LAC continues to select Tunnel 1 until the destination session count for 192.168.1.1 equals 40, matching the count for 192.168.2.2 in Tunnel 2.
2. When the next subscriber logs in, the LAC randomly selects between Tunnel 1 and Tunnel 2, because their destinations have the same session count, and it is lower than that for Tunnel 3 (300).
3. Whichever tunnel is selected from this pair, the session count for its destination is now 41. The other tunnel is selected when the next subscriber logs in, because it has the lower destination session count of 40. This raises its destination session count to 41, matching the other tunnel.
4. As subscribers continue to log in, the LAC repeats this process, randomly selecting between Tunnel 1 and Tunnel 2 when their session counts match and then selecting the other tunnel with the next subscriber, until their destination session counts both reach 200, matching their tunnel maximum session limit of 200. Because both tunnels have reached their maximum session limit, they are not available for selection.

5. As subscribers continue to log in, the LAC selects the remaining tunnel in the preference level, Tunnel 3, until the session count for its destination reaches the maximum session limit for the tunnel, 1000.
6. When the next subscriber logs in, the LAC drops to the next preference level and selects Tunnel 4, because it is the only tunnel at this level.
7. As subscribers continue to log in, the LAC continues to select Tunnel 4, because no maximum session limit is configured for this tunnel. The LAC can subsequently select a tunnel in the higher preference level only when a session is terminated for one of the tunnels at that level, dropping its session count below the maximum limit.

In Scenario 5, one of the destinations is locked:

- Tunnel 1, preference level 1, 192.168.1.1, destination session count = 100, destination locked out
- Tunnel 2, preference level 1, 192.168.2.2, destination session count = 200
- Tunnel 3, preference level 1, 192.168.3.3, destination session count = 250

When the first PPP user tries to connect to the domain, the LAC cannot select Tunnel 1, even though its destination has the lowest session count, because the tunnel is in the destination lockout state. Tunnel 1 cannot be considered until it is out of the locked state. The LAC selects Tunnel 2 because the session count for 192.168.2.2 is lower than for 192.168.3.3.

When additional PPP users try to connect to the domain, what happens next depends on when 192.168.1.1 emerges from the lockout state. For as long as 192.168.1.1 is locked out, the LAC makes the selections as follows:

1. The LAC continues to select Tunnel 2 until the session count for 192.168.2.2 equals 250, matching the count for 192.168.3.3 in Tunnel 3.
2. When the next subscriber logs in, the LAC randomly selects between Tunnel 2 and Tunnel 3, because their destinations have the same session count, 250.
3. Whichever tunnel is selected from this pair, the session count for its destination is now 251. The other tunnel is selected when the next subscriber logs in, because it has the lower destination session count of 250. This raises its destination session count to 251, matching the other tunnel.
4. As subscribers continue to log in, the LAC repeats this process, randomly selecting between Tunnel 2 and Tunnel 3 when their session counts match and then selecting the other tunnel with the next subscriber.

Whenever 192.168.1.1 emerges from the lockout state, the LAC selects Tunnel 1 for the next subscriber because 192.168.1.1 has the lowest session count. The LAC continues to do so until the session count for 192.168.1.1 matches the current session count for either of the other destinations. From that point forward, the LAC alternates making a random selection between tunnels with matching destination session counts and then subsequently selecting the tunnel with the lowest count.

Whenever 192.168.1.1 emerges from the lockout state,

1. The LAC selects Tunnel 1 for the next subscriber because 192.168.1.1 has the lowest session count.
2. The LAC continues to select Tunnel 1 until the session count for 192.168.1.1 matches the current session count for either of the other destinations.
3. From that point forward, the LAC alternates making a random selection between tunnels with matching destination session counts and then subsequently selecting the tunnel with the lowest count.

L2TP Session Limits Overview

IN THIS SECTION

- Scenario 1: Chassis Limit | 201
- Scenario 2: Tunnel Limit | 201
- Scenario 3: Tunnel Group Limit | 202
- Scenario 4: Session-Limit Group Limit | 203
- Scenario 5: Individual Client Limit | 205

When an L2TP session request is initiated, the LNS or LAC checks the number of current active sessions against the maximum number of sessions allowed for the chassis, tunnels, a tunnel group, a client (requesting host device), or a group of clients. New session requests are rejected when the configured session limit is reached.

When a session is requested, the LNS checks for session limits in the following order:

chassis > tunnel > tunnel group > session-limit group > client

At each level, the LNS determines whether the current session count is less than the configured limit. When that is true or when no limit is configured, the check passes and the LNS proceeds to check the next level. If at any level the current session count is equal to the configured limit, then the LNS rejects the session request and does not check any other level. Otherwise, the session can be established.

When a session request is rejected for an existing tunnel, a Call-Disconnect-Notify (CDN) message with a result code and error code both set to 4 is returned in response to the incoming-call request (ICRQ). When the rejected request is for a new tunnel, the tunnel is established but the session fails to come up, causing the tunnel to come down because it has no sessions.

The LAC performs the same check, but only for the chassis and tunnel levels. The LAC rejects requests by returning a PPP terminate message to the client.

You can configure session limits for the chassis, all tunnels, a tunnel group, a group of clients, or an individual client. The scenarios that follow describe what happens for different configurations of session limits.

Scenario 1: Chassis Limit

In [Table 15 on page 201](#), the current L2TP session count is 10,000 and the session limit is configured as 10,000 at every level. When a new session is requested, the first check at the chassis level fails, because the current session count matches the configured limit. No further checks are performed at the other levels and the session request is rejected. No new sessions are allowed at any level until the current session count drops below 10,000.

Table 15: Scenario 1, Chassis Limit

Level	Configured Session Limit	Current Session Count Displayed by show services l2tp summary Command	Session Limit Check Result
Chassis	10,000	10,000	Fail
Tunnel A	10,000	10,000	-
Tunnel group B	10,000	10,000	-
Session-limit group	10,000	10,000	-
Client	10,000	10,000	-

Scenario 2: Tunnel Limit

In [Table 16 on page 202](#), the current L2TP session count is 2000. When a new session is requested, the first check at the chassis level passes because the configured limit allows up to 10,000 sessions on the chassis, but only 2000 sessions are currently active. The next check, at the tunnel level, fails, because the current session count matches the configured limit tunnel limit of 2000 for tunnel A.

No further checks are performed at the other levels and the session request is rejected.

Table 16: Scenario 2, Tunnel Limit

Level	Configured Session Limit	Current Session Count Displayed by show services l2tp summary Command	Session Limit Check Result
Chassis	10,000	2000	Pass
Tunnel A	2000	2000	Fail
Tunnel group B	10,000	2000	-
Session-limit group	6000	2000	-
Client	6000	2000	-

No new sessions are allowed on tunnel A until its current session count drops below 2000 and the session check can pass. If that happens, then the other level checks pass in this scenario because their configured limits are greater than their current counts.

The session limit of 2000 applies to all tunnels; that is, each active tunnel has an independent limit of 2000 sessions. The failure of one tunnel has no effect on other tunnels. A session request on any other tunnel passes, as long as the current session count for that tunnel is less than 2000.

Scenario 3: Tunnel Group Limit

In [Table 17 on page 203](#), the current L2TP session count is 2000. When a new session is requested, the first check at the chassis level passes because the configured limit allows up to 10,000 sessions on the chassis, but only 2000 sessions are currently active. The second check, at the tunnel level, also passes for the same reason. The next check, at the tunnel group level for tunnel group B, fails, because the current session count for tunnel group B matches the configured limit tunnel group limit of 2000.

No further checks are performed at the other levels and the session request is rejected.

Table 17: Scenario 3, Tunnel Group Limit

Level	Configured Session Limit	Current Session Count Displayed by show services l2tp summary Command	Session Limit Check Result
Chassis	10,000	2000	Pass
Tunnel A	10,000	2000	Pass
Tunnel group B	2000	2000	Fail
Session-limit group	6000	2000	-
Client	6000	2000	-

No new sessions are allowed on tunnel group B until its current session count drops below 2000 and the session check can pass. If that happens, then the other level checks can pass because their configured limits are greater than their current counts.

For tunnel groups, the session limit is configured on a per-group basis; that is, you cannot specify a single limit that applies to all tunnel groups. The failure of any tunnel group has no effect on other tunnel groups. In this scenario, a session request on any other tunnel group passes, if the current session count for that group is less than its configured session limit.

Scenario 4: Session-Limit Group Limit

In [Table 18 on page 204](#), the current L2TP session count is 6000. When a new session is requested, the check passes for the chassis, tunnel, and tunnel group because the configured limit for each allows up to 10,000 sessions, but only 6000 sessions are currently active. The check at the session-limit group fails, because the current session count for session-limit group slg1 matches the configured limit of 6000.

No further checks are performed at the remaining level and the session request is rejected.

Table 18: Scenario 4, Session-Limit Group Limit

Level	Configured Session Limit	Current Session Count Displayed by show services l2tp summary Command	Session Limit Check Result
Chassis	10,000	6000	Pass
Tunnel A	10,000	6000	Pass
Tunnel group B	10,000	6000	Pass
Session-Limit group slg1	6000	6000	Fail
Client	8000	2000	-

No new sessions are allowed for any clients in session-limit group slg1 until the group's current session count drops below 6000 and the session check can pass. If that happens, then the remaining level check can pass because its configured limit is greater than its current count.

You can reconfigure a session-limit group by removing or adding clients without affecting any current sessions. The reconfiguration does affect the number of sessions available to be established for the client group.

- If you remove a client, then the number of new sessions that can be established increases by the number of that client's current sessions.
- If you add a client, then the number of new sessions that can be established is reduced by the number of that client's current sessions. The new total of current sessions for existing clients plus the new client can exceed the configured limit for the session-limit group. In this case, no sessions are dropped, but no new sessions can be established until the session count drops below the configured group limit.

To explore this further, consider the following sequence of events:

1. The session-limit group slg1 has two clients, ent1-serviceA with a current session count of 3500 and ent1-serviceB with a current session count of 0. Because group slg1 has a limit of 6000, no more than 2500 sessions can be added for these clients:

$$6000 - 3500 = 2500$$

2. Then 1000 sessions are logged in for client ent1-service B. Now no more than 1500 sessions can be added for these clients:

$$6000 - (3500 + 1000) = 1500$$

3. Next, suppose you remove client ent1-serviceA from the session-limit group. The group session capacity increases to 5000 sessions:

$$6000 - 1000 = 5000$$

4. Finally, you add a new client, ent1-serviceC, to the session-limit group. This new client currently has 8000 active sessions. In this case, the session-limit group now has 9000 sessions:

$$1000 + 8000 = 9000$$

No sessions are dropped even though the maximum session limit for the group, 6000, is exceeded. No new sessions can be added until the session count drops from 9000 to below 6000.

Scenario 5: Individual Client Limit

In [Table 19 on page 205](#), the session check passes for the chassis, tunnel, and tunnel group because their configured limits are greater than their current session counts. The client, ent1-serviceA, does not belong to a session-limit-group. The limit check fails for the client because its current session count matches the configured limit of 6000.

Table 19: Scenario 5, Individual Client Limit

Level	Configured Session Limit	Current Session Count Displayed by show services l2tp summary Command	Session Limit Check Result
Chassis	10,000	6000	Pass
Tunnel A	10,000	6000	Pass
Tunnel group B	8000	6000	Pass
Client ent1-serviceA	6000	6000	Fail

No new sessions are allowed for this client until its current session count drops below 6000 and the session check can pass. The failure of any independent client has no effect on other clients. In this scenario, a session request for any other independent client passes, if the current session count for that client is less than its configured session limit.

The session limit that you set for an individual client—one that is not part of a session-limit group—applies on a per-tunnel-group basis. Multiple LACs with the same source hostname but different source IP addresses are treated as the same client.

Suppose you have three LACs, A, B, and C. All three have the same source hostname, ce-lac. LAC A and LAC B establish sessions with an LNS through the gateway address associated with tunnel group 1. LAC C establishes sessions through a different gateway associated with tunnel group 2. Because the LACs have the same hostname, the client configuration is the same for all three. However, the client session limit applies differently to the LACs because of the tunnel groups.

Suppose the client session limit is 100. Because LAC A and LAC B both create sessions in tunnel group 1, they must share the client limit. That means that the total number of sessions allowed for LAC A and LAC B combined is 100.

LAC C creates sessions in a different tunnel group, 2. Because the client session limit applies per tunnel group, then LAC C is allowed 100 sessions, regardless of how many sessions LAC A and LAC B have already established.

Limiting the Number of L2TP Sessions Allowed by the LAC or LNS

You can place a limit on the maximum number of L2TP sessions allowed for the chassis, all tunnels, a tunnel group, a group of clients, an individual client, or an individual service interface or aggregated service interface. New session requests are rejected by the LNS or LAC when the configured session limit is reached. Session requests are also rejected when the maximum chassis limit has been reached, even when a configured limit is not exceeded. Configurable session limits provide fine-grained control of the number of sessions that a customer can have while connected over LACs in multiple locations.

NOTE: You cannot set the limit to be more than the default maximum limit for the chassis.

To limit the number of sessions allowed on a chassis (LAC or LNS):

- Configure the maximum number of sessions.

```
[edit services l2tp]
user@host# set maximum-sessions number
```

To limit the number of sessions per tunnel for all tunnels (LAC or LNS):

- Configure the maximum number of sessions.

```
[edit services l2tp tunnel ]
user@host# set maximum-sessions number
```

You cannot set the limit to be more than 65,535 sessions.

To limit the number of sessions for all tunnels in a specific tunnel group (LNS):

- Configure the maximum number of sessions.

```
[edit services l2tp tunnel-group tunnel-group-name]
user@host# set maximum-sessions number
```

To limit the number of sessions that are allowed on an individual service interface:

- Configure the maximum number of sessions.

```
[edit interfaces si-slot/pic/port]
user@host# set l2tp-maximum-session number
```

To limit the number of sessions that are allowed on an individual aggregated service interface:

- Configure the maximum number of sessions.

```
[edit interfaces asinumber]
user@host# set l2tp-maximum-session number
```

NOTE: The configuration applies to all member interfaces; the limit cannot be configured for individual member interfaces of the aggregated service interface.

To limit the number of sessions for a group of clients (LNS):

1. Configure the maximum number of sessions.

```
[edit services l2tp sessions-limit-group limit-group-name]
user@host# set maximum-sessions number
```

2. Associate a client with the session-limit group.

```
[edit access profile profile-name client client-name l2tp]
user@host# set sessions-limit-group limit-group-name
```

To limit the number of sessions for a client that is not a member of a session-limit group (LNS):

- Configure the maximum number of sessions.

```
[edit access profile profile-name client client-name]
user@host# set maximum-sessions number
```

NOTE: Configuring the session limit at any level to be less than the number of sessions that currently exist at that level has no effect on existing sessions. The new limit applies only if the number of sessions drops below the new limit.

You can use the `show services l2tp summary extensive` command to display the configured sessions limit for a tunnel:

```
user@host> show services l2tp tunnel extensive
...
  Max sessions: 32000, Window size: 4, Hello interval: 60
...
```

The displayed limit for configured sessions is set to the lowest of the following configured session values:

- Global (chassis)—(LAC and LNS) `set services l2tp tunnel maximum-sessions number`
- Tunnel profile (individual tunnel)—(LAC and LNS) `set access tunnel-profile profile-name tunnel tunnel-id maximum-sessions number`
- RADIUS—(LAC and LNS) Value of VSA 26–33, Tunnel-Max-Sessions
- Host profile—(LNS only) `set access profile profile-name client client-name l2tp maximum-sessions-per-tunnel`

The configured values determine the field value starting in the following Junos OS releases: 19.2R3, 19.3R3, 19.4R3, 20.1R2, 20.2R2, and 20.3R1. In earlier releases, the field displays the host profile value for the LNS, but it displays a fixed value of 512,000 for the LAC.

NOTE: After a GRES, a unified ISSU, or a restart of the jl2tpd process, the value of this field is accurate only after a new session comes up on the tunnel. Until that happens, the field shows a value of 65,535 instead of the configured value.

Suppose you have two tunnels, tunnel A and tunnel B. A GRES takes place, and the field for each tunnel shows 65,535. When a new session comes up on tunnel B, the value for that tunnel updates to the configured value. For tunnel A, the field continues to show 65,535 until that tunnel gets a new session.

Setting the Format for the Tunnel Name

By default, the name of a tunnel corresponds to the Tunnel-Assignment-Id [82] returned by the AAA server. You can optionally configure the LAC to use more elements in the construction of a tunnel name by including the `assignment-id-format client-server-id` statement at the `[edit services l2tp tunnel]` hierarchy level. This format uses three attributes: Tunnel-Client-Auth-Id [90], Tunnel-Server-Endpoint [67], and Tunnel-Assignment-Id [82]. These attributes correspond, respectively, to the values configured in the tunnel profile for the LAC (source gateway) name, the tunnel endpoint (remote gateway) address on the LNS, and the tunnel ID.

A consequence of the `client-server-id` format is that the LAC automatically creates a new tunnel when the AAA server returns a different Tunnel-Client-Auth-Id than previously returned.

NOTE: Before you downgrade to a Junos OS Release that does not support this statement, we recommend that you explicitly unconfigure the feature by including the `no assignment-id-format assignment-id` statement at the `[edit services l2tp tunnel]` hierarchy level.

To change how the tunnel name is formatted:

- Configure the format.

```
[edit services l2tp tunnel]
user@host# set assignment-id-format client-server-id
```

Configuring a Tunnel Profile for Subscriber Access

The tunnel profile specifies a set of attributes to characterize the tunnel. The profile can be applied by a domain map or automatically when the tunnel is created.

NOTE: RADIUS attributes and VSAs can override the values you configured by a tunnel profile in a domain map. In the absence of a domain map, RADIUS can supply all the characteristics of a tunnel. The steps in the following procedure list the corresponding standard RADIUS attribute or VSA that you can configure on your RADIUS server to modify or configure the tunnel profile. RADIUS-supplied attributes are associated with a tunnel by a tag carried in the attribute, which matches the tunnel identifier. A tag of 0 indicates the tag is not used. If L2TP receives a RADIUS attribute with a tag of 0, the attribute cannot be merged with the tunnel profile configuration corresponding to the subscriber domain because a tunnel profile cannot provide a tunnel tag (tunnel identifier) of 0. Only tags in the range of 1 through 31 are supported.

To configure a tunnel definition for a tunnel profile:

1. Specify the tunnel profile for which you are defining a tunnel. (Tunnel-Group [26-64])

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

2. Specify an identifier (name) for the L2TP control connection for the tunnel.

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

3. Configure the IP address of the local L2TP tunnel endpoint, the LAC. (Tunnel-Client-Endpoint [66])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set source-gateway address client-ip-address
```

4. Configure the IP address of the remote L2TP tunnel endpoint, the LNS. (Tunnel-Server-Endpoint [67])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set remote-gateway address server-ip-address
```

5. (Optional) Configure the preference level for the tunnel. (Tunnel-Preference [83])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set preference number
```

6. (Optional) Configure the hostname of the local client (LAC). (Tunnel-Client-Auth-Id [90])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set source-gateway gateway-name client-name
```

7. (Optional) Configure the hostname of the remote server (LNS). (Tunnel-Server-Auth-Id [91])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set remote-gateway gateway-name server-name
```

8. (Optional) Specify the medium (network) type for the tunnel. (Tunnel-Medium-Type [65])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set medium type
```

9. (Optional) Specify the protocol type for the tunnel. (Tunnel-Type [64])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set type tunnel-type
```

10. (Optional) Configure the assignment ID for the tunnel. (Tunnel-Assignment-Id [82])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set identification name
```

11. (Optional) Configure the maximum number of sessions allowed in the tunnel. (Tunnel-Max-Sessions [26-33])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]  
user@host# set max-sessions number
```

12. (Optional) Configure the password for remote server authentication. (Standard RADIUS attribute Tunnel-Password [69] or VSA Tunnel-Password [26-9])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set secret password
```

13. (Optional) Configure the logical system to use for the tunnel.
If you configure a logical system, you must also configure a routing instance.

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set logical-system logical-system-name
```

14. (Optional) Configure the routing instance to use for the tunnel. (Tunnel-Virtual-Router [26-8])
If you configure a routing instance, configuring a logical system is optional.

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set routing-instance routing-instance-name
```

15. (Optional) Enable the LAC to interoperate with Cisco LNS devices. (Tunnel-Nas-Port-Method [26-30])

```
[edit access tunnel-profile profile-name tunnel tunnel-id]
user@host# set nas-port-method cisco-avp
```

The following example shows a complete configuration for a tunnel profile:

```
tunnel-profile marketing {
  tunnel 1 {
    preference 5;
    remote-gateway {
      address 198.51.100.4;
      gateway-name work;
    }
    source-gateway {
      address 192.0.2.10;
      gateway-name local;
    }
    secret $ABC123;
    logical-system bos-metro-5;
```



```
routing-instance rox-12-32;  
medium ipv4;  
type l2tp;  
identification tunnel_to_work;  
max-sessions 32;  
nas-port-method cisco avp;  
}  
}
```

Configuring the L2TP LAC Tunnel Selection Parameters

When the LAC determines that a PPP session should be tunneled, it selects a tunnel from the set of tunnels associated with either the PPP user or the PPP user's domain. You can configure how a tunnel is selected and whether certain information is sent by the LAC to the LNS.

To configure tunnel selection parameters:

1. (Optional) Configure how a tunnel is selected when a connection attempt fails.
See ["Configuring LAC Tunnel Selection Failover Within a Preference Level"](#) on page 213.
2. (Optional) Configure how sessions are load-balanced among tunnels.
See ["Configuring Weighted Load Balancing for LAC Tunnel Sessions"](#) on page 214.
3. (Optional) Configure sessions to be load-balanced among tunnels within a preference level, by distributing the sessions equally among all tunnels.
See ["Configuring Destination-Equal Load Balancing for LAC Tunnel Sessions"](#) on page 215.

Configuring LAC Tunnel Selection Failover Within a Preference Level

You can configure how LAC tunnel selection continues in the event of a failure to connect. By default, when the router is unable to connect to a destination at a given preference level, it attempts to connect at the next lower level. You can specify that the router instead attempt to connect to another destination at the same level as the failed attempt.

If all destinations at a preference level are marked as unreachable, the router does not attempt to connect to a destination at that level. It drops to the next lower preference level to select a destination.

If all destinations at all preference levels are marked as unreachable, the router chooses the destination that failed first and tries to make a connection. If the connection fails, the router rejects the PPP user session without attempting to contact the remote router.

For example, suppose there are four tunnels for a domain: A, B, C, and D. All tunnels are considered reachable, and the preference levels are assigned as follows:

- A and B at preference 0
- C and D at preference 1

When the router attempts to connect to the domain, suppose it randomly selects tunnel B from preference 0. If it fails to connect to tunnel B, the router excludes tunnel B for five minutes and attempts to connect to tunnel A. If this attempt also fails, the router drops to preference 1. Then suppose the router selects tunnel C. If it also fails to connect to tunnel C, the router excludes tunnel C for five minutes and attempts to connect to tunnel D.

You configure the preference level used for this tunnel selection method in the tunnel profile or the RADIUS Tunnel-Preference [83] attribute.

To enable tunnel selection failover within a preference level:

- Specify failover within preference.

```
[edit services l2tp]
user@host# set failover-within-preference
```

Configuring Weighted Load Balancing for LAC Tunnel Sessions

By default, the L2TP LAC selects tunnels for new sessions at random from within the highest available preference level. You can configure the LAC to distribute sessions across tunnels at the highest available preference level by evaluating the weight of each tunnel. This method is known as *weighted load balancing*. The weight of a tunnel is proportional to its maximum session limit and the maximum session limits of the other tunnels at the same preference level. When you configure weighted load balancing, the LAC still selects tunnels randomly within a preference level, but on average the sessions are distributed across tunnels in relationship to the tunnel weights.

To configure weighted load balancing:

- Specify load balancing.

```
[edit services l2tp]
user@host# set weighted-load-balancing
```

Configuring Destination-Equal Load Balancing for LAC Tunnel Sessions

By default, the L2TP LAC selects tunnels for new sessions at random from within the highest available preference level. Starting in Junos OS Release 15.1, you can configure the LAC to distribute sessions equally across all tunnels at the highest available preference level by evaluating the number of sessions to the destinations and the number of sessions carried by the tunnels. This distribution method is known as *destination-equal load balancing*. The LAC selects the tunnel with the lightest load, according to the following guidelines:

- When each tunnel goes to a separate destination and only one destination has the lowest session count among all destinations, the LAC selects the tunnel to that destination.
- When each tunnel goes to a separate destination and more than one destination has the same lowest session count, the LAC selects a tunnel at random from among the tunnels to these destinations.
- When more than one tunnel goes to the same destination and that destination has the lowest destination session count, the LAC selects from among these tunnels the one that has the lowest total number of tunnel sessions. If the tunnel session count is the same for all these tunnels, then the LAC selects one of them at random.

To configure destination-equal load balancing:

- Specify destination-equal load balancing.

```
[edit services l2tp]
user@host# set destination-equal-load-balancing
```

Enabling the LAC for IPv6 Services

You can configure the LAC to create the IPv6 address family (inet6) when tunneling the subscribers to the LNS. IPv6 firewall filters can then be applied by services on the LAC to subscriber traffic. By default, the LAC requires only family inet to enable forwarding into an IP tunnel. The LAC can apply IPv4 firewall filters to the session. Even when family inet6 is included in the dynamic profile, by default it is not created in order to conserve resources, because it is not needed. Consequently IPv6 firewall filters cannot be applied.

To enable IPv6 address family creation and the application of IPv6 firewall filters:

- Configure enabling.

```
[edit services l2tp]
user@host# set enable-ipv6-services-for-lac
```

You can use the `show services l2tp summary` command to display whether the statement is enabled or disabled.

Testing L2TP Tunnel Configurations from the LAC

You can test L2TP tunnel configurations on the LAC and successful subscriber authentication and tunneling without bringing up a PPP user and an associated tunnel.

Issue the `test services l2tp tunnel` command from CLI operational mode to map a subscriber to an L2TP tunnel, verify the L2TP tunnel configuration (both locally on the LAC and on a back-end server such as a RADIUS server), and verify that L2TP tunnels from the LAC can be established with the remote LNS.

The Junos OS LAC implementation enables you to configure multiple tunnels from which one tunnel is chosen for tunneling a PPP subscriber. You can use the `test services l2tp tunnel` command to test all possible tunnel configurations to verify that each can be established. Alternatively, you can test only a specific tunnel for the subscriber.

You must specify a configured subscriber username when you issue the command. The test generates a dummy password—*testpass*—for the subscriber, or you can optionally specify the password. The test verifies whether the subscriber identified by that username can be tunneled according to the tunnel configuration. If the subscriber can be tunneled, then the test verifies whether the L2TP tunnel can be established with the LNS according to the L2TP configuration.

You can optionally specify a tunnel ID, in which case only that tunnel is tested; the tunnel must be already configured for that username. If you omit this option, the test is applied to the full set of tunnel configurations that are returned for the username. The tunnel ID you specify is the same as that used by Tunnel-Assignment-Id (RADIUS attribute 82) and specified by the `identification` statement in the tunnel profile.

To test subscriber authentication and tunnel configuration:

- Specify only the username.

Example 1:

```
user@host> test services l2tp tunnel user test-user1@example.com
Subscriber: test-user1@example.com, authentication failed
```

The user failed authentication with the generated password and consequently was not tunneled.

Example 2:

```
user@host> test services l2tp tunnel user user23@example.com
Subscriber: user23@example.com, authentication success, l2tp tunneled
Tunnel-name  Tunnel-peer  Logical-System  Routing-Instance  Status
test1tunnel  192.168.2.3  default         default            Up
test2tunnel  192.168.30.3  default         default            Peer unresponsive
test3tunnel  192.168.50.1  default         test               Up
```

This user was authenticated with the generated password and successfully tunneled. A set of tunnels was found to be associated with that username and the entire set was tested.

- Specify the username and the user's configured password.

```
user@host> test services l2tp tunnel user test-user1@example.com password grZ98#jW
Subscriber: test-user1@example.com, authentication success, locally terminated
```

The subscriber was authenticated. However, the user was terminated locally rather than tunneled; this means that no tunnel was found to be associated with the user.

- Specify the username and a particular tunnel for the subscriber.

```
user@host> test services l2tp tunnel user rx37w@example.com tunnel-name ce-lac
Subscriber: rx37w@example.com, authentication success, l2tp tunneled
Tunnel-name  Tunnel-peer  Logical-System  Routing-Instance  Status
ce-lac       192.168.5.10  default         default            Up
```

The subscriber was authenticated and tunneled. The specified tunnel was found for the subscriber and the tunnel was established, confirming the tunnel configuration.

- Specify the username, the user's configured password, and a tunnel.

```
user@host> test services l2tp tunnel user fanta4-mfg-fan@example.com password dieda499 tunnel-
name tunnel5
Subscriber: fanta4-mfg-fan@example.com, authentication success, l2tp tunneled
```

The subscriber was authenticated and tunneled. The absence of tunnel information in the output indicates that the specified tunnel configuration does not exist.

Release History Table

Release	Description
20.3R1	The configured values determine the field value starting in the following Junos OS releases: 19.2R3, 19.3R3, 19.4R3, 20.1R2, 20.2R2, and 20.3R1.
15.1	Starting in Junos OS Release 15.1, you can configure the LAC to distribute sessions equally across all tunnels at the highest available preference level by evaluating the number of sessions to the destinations and the number of sessions carried by the tunnels.

RELATED DOCUMENTATION

[L2TP for Subscriber Access Overview | 144](#)

[Configuring an L2TP LAC | 176](#)

[Configuring an L2TP LNS with Inline Service Interfaces | 259](#)

[Configuring the L2TP Destination Lockout Timeout | 172](#)

Specifying a Tunnel Profile in a Domain Map

[L2TP Session Limits and Load Balancing for Service Interfaces | 282](#)

[Configuring L2TP Tunnel Groups](#)

[Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces | 301](#)

[Configuring 1:1 LNS Stateful Redundancy on Aggregated Inline Service Interfaces | 276](#)

Domain Mapping Overview

[LAC Interoperation with Third-Party LNS Devices | 180](#)

Filtering RADIUS Attributes and VSAs from RADIUS Messages

L2TP Subscriber Access Lines and Connection Speeds

IN THIS SECTION

- [Subscriber Access Line Information Handling by the LAC and LNS Overview | 219](#)
- [Transmission of Tx and Rx Connection Speeds from LAC to LNS | 232](#)
- [Transmission of the Receive Connect Speed AVP When Transmit and Receive Connect Speeds are Equal | 241](#)
- [Configuring the Method to Derive the LAC Connection Speeds Sent to the LNS | 242](#)
- [Configuring the Reporting and Processing of Subscriber Access Line Information | 245](#)
- [Preventing the LAC from Sending Calling Number AVP 22 to the LNS | 250](#)
- [Override the Calling-Station-ID Format for the Calling Number AVP | 251](#)
- [Specifying a Rate-Limiting Service Profile for L2TP Connection Speeds | 253](#)

Subscriber Access Line Information Handling by the LAC and LNS Overview

IN THIS SECTION

- [Access Line Information Forwarding | 220](#)
- [Access Line Information AVPs | 221](#)
- [Connection Speed Updates on the LAC | 230](#)
- [Connection Speed Updates on the LNS | 231](#)
- [Interaction Between Global and Per-Destination Configurations | 231](#)

Starting in Junos OS Release 14.1, L2TP supports a set of AVPs that convey information about subscriber access lines from the LAC to the LNS. The information originates from an ANCP access node (DSLAM) and is distributed to the LAC by means of either DSL Forum VSAs in ANCP messages or PPPoE

intermediate agent tags included in the PPPoE PADI and PADR messages. The access node is typically a DSLAM for DSL access networks or, starting in Junos OS Release 19.3R1, an ONT/ONU for PON access networks. See the following references for more information about DSL Forum VSAs and L2TP AVPs:

- RFC 4679, *DSL Forum Vendor-Specific RADIUS Attributes*
- RFC 5515, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*
- RFC 6320, *Protocol for Access Node Control Mechanism in Broadband Networks*
- RFC 6320 Draft Extension, *Access Extensions for the Access Node Control Protocol*
- Broadband Forum technical report TR-101, *Migration to Ethernet-Based Broadband Aggregation*

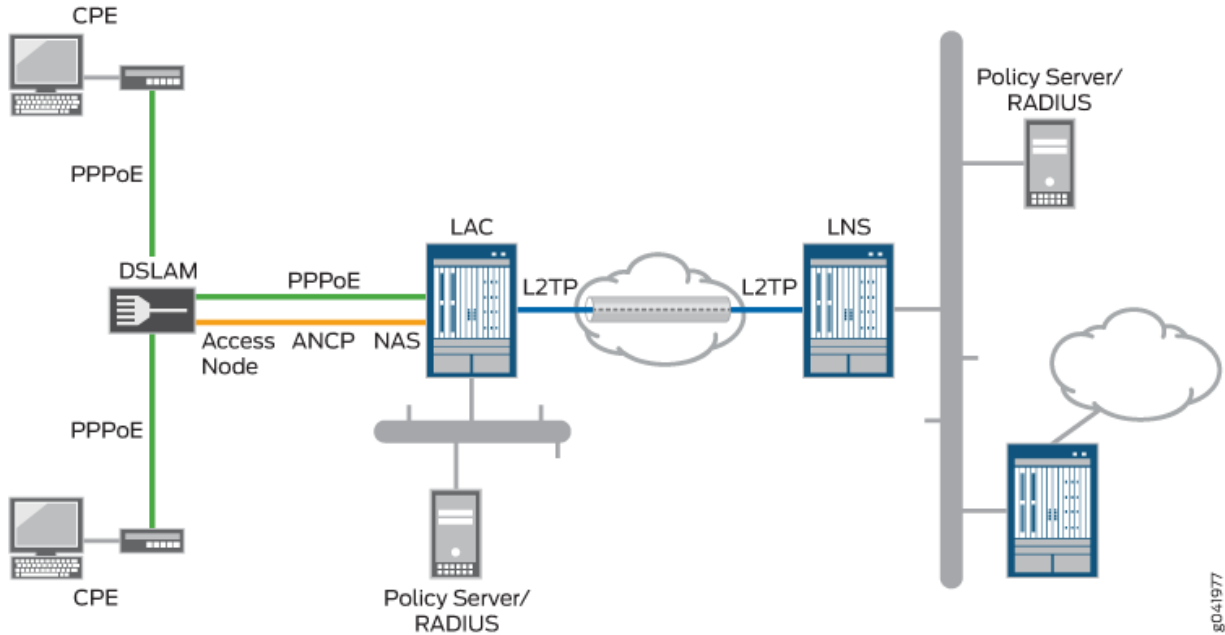
Access Line Information Forwarding

In the network topology shown in [Figure 16 on page 221](#), when a subscriber initiates a connection through the CPE, the DSLAM relays the subscriber's PPPoE session to the router configured as a LAC. When the router has established the PPPoE session, the LAC initiates an L2TP tunnel to forward the subscriber's encapsulated PPP packets into the provider network.

In parallel to the PPPoE session, an ANCP connection between the DSLAM and the ANCP agent on the router conveys information about the subscriber's local loop as well as the link speeds of the PPPoE sessions on the local loop. The DSLAM sends the router Agent Circuit Id (ACI) and Agent Remote Id (ARI) strings that uniquely identify the DSLAM's receiving interface; this information is encoded in the ANCP Port Up and Port Down messages as Access Line Identifying TLVs. The ANCP messages can also include line attributes such as minimum, maximum, and actual net upstream and downstream data rates in the DSL Line Attributes TLV. The DSLAM can also send the access line attributes in vendor-specific tags that it inserts in the PADI and PADR messages.

NOTE: Starting in Junos OS Release 19.3R1, the access nodes for PON subscriber access lines (such as ONTs and ONUs) are supported in this same scenario, in addition to the previously supported DSL access nodes.

Figure 16: Sample L2TP Network Topology



Access Line Information AVPs

L2TP supports the AVPs listed in [Table 20 on page 222](#) to carry this information. The access line information is not required for the L2TP session to be initiated, and the establishment of that session is not delayed waiting for the values to be sent from the access node. The content of the ICRQ message generally varies between DSL access lines and PON access lines. AVPs, 1, 2, 3, and 6 are used for access line identification for both DSL and PON. If PON information is reported using DSL AVPs, then the content is the same as it would be for DSL access.

The access line information provided by the AVPs in ICRQ messages is passed on to RADIUS in DSL Forum VSAs. It is not used for shaping the traffic rate on the subscriber access lines.

Table 20: L2TP AVPs That Provide Subscriber Access Line Information

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
1 Agent-Circuit-Id	Identifier for the subscriber agent circuit ID (ACI) that corresponds to the access node interface from which subscriber requests are initiated. 2-63 octet string	ICRQ DSL, PON	26-3561-1
2 Agent-Remote-Id	Unique identifier for the subscriber associated with the access node interface from which requests are initiated. 2-63 octet string	ICRQ DSL, PON	26-3561-2
3 Access-Aggregation-Circuit-ID-ASCII	ASCII identifier for the subscriber access line, based on its network-facing logical appearance If the string begins with a # sign, then the remainder of the string represents a logical intermediate node (DPU-C or PON tree) in the access network to which the subscriber is attached. The string is used as the name of a CoS Level 2 interface set that groups subscribers.	ICRQ DSL, PON	26-3561-3

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
6 Access-Aggregation- Circuit-ID-Binary	Binary identifier for the subscriber access line 32-bit or 64-bit string	ICRQ DSL, PON	26-3561-6
97 Connect-Speed-Update	Data structure listing remote session id and the current transmit and receive connection speeds in bits per second.	CSUN, CSURQ	(none)
98 Connect-Speed-Update- Enable	Value does not matter: presence indicates support for CSUN, CSURQ message types for this session.	ICRQ	(none)
129 Actual-Data-Rate- Upstream	Actual upstream data rate of the subscriber's synchronized DSL link, in bps 64-bit unsigned integer; data rate in bits per sec	ICRQ DSL	26-3561-129
130 Actual-Data-Rate- Downstream	Actual downstream data rate of the subscriber's synchronized DSL link, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-130
131 Minimum-Data-Rate- Upstream	Minimum upstream data rate configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-131

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
132 Minimum-Data-Rate-Downstream	Minimum downstream data rate configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-132
133 Attainable-Data-Rate-Upstream	Upstream data rate that the subscriber can attain, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-133
134 Attainable-Data-Rate-Downstream	Downstream data rate that the subscriber can attain, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-134
135 Maximum-Data-Rate-Upstream	Maximum upstream data rate configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-135
136 Maximum-Data-Rate-Downstream	Maximum downstream data rate configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-136
137 Minimum-Data-Rate-Upstream-Low-Power	Minimum upstream data rate in low power state configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-137

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
138 Minimum-Data-Rate-Downstream-Low-Power	Minimum downstream data rate in low power state configured for the subscriber, in bps 64-bit unsigned integer	ICRQ DSL	26-3561-138
139 Maximum-Interleaving-Delay-Upstream	Maximum one-way upstream interleaving delay configured for the subscriber, in milliseconds 32-bit unsigned integer	ICRQ DSL	26-3561-139
140 Actual-Interleaving-Delay-Upstream	Subscriber's actual one-way upstream interleaving delay, in milliseconds 32-bit unsigned integer	ICRQ DSL	26-3561-140
141 Maximum-Interleaving-Delay-Downstream	Maximum one-way downstream interleaving delay configured for the subscriber, in milliseconds 32-bit unsigned integer	ICRQ DSL	26-3561-141
142 Actual-Interleaving-Delay-Downstream	Subscriber's actual one-way downstream interleaving delay, in milliseconds 32-bit unsigned integer	ICRQ DSL	26-3561-142

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
144 Access-Loop-Encapsulation	Encapsulation used by the subscriber associated with the access node interface from which requests are initiated Three one-octet encodings for data link, encapsulation 1, and encapsulation 2.	ICRQ DSL	26-3561-144
145 ANCP-Access-Line-Type (This corresponds to the ANCP DSL-Type TLV.)	One octet encoding for transmission system type, followed by three MBZ (must be zero) octets (total 4 bytes). This value is not supplied in the ICRQ when the access line parameters are sourced from PPPoE-IA, because the ANCP-sourced information may not be immediately available. Starting in Junos OS Release 18.1R1, this AVP is included even when the line type is 0 for OTHER access line types.	ICRQ DSL	26-3561-145

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
146 PON-Access-Type	Type of PON access line in use: <ul style="list-style-type: none"> • 0—OTHER • 1—GPON • 2—XG-PON1 • 3—TWDM-PON • 4—XGS-PON • 5—WDM-PON • 7—UNKNOWN 32-bit unsigned integer	ICRQ PON	26-3561-146
147 ONT/ONU-Average-Data-Rate-Downstream	Average downstream data rate for ONT/ONU, in bps 64-bit unsigned integer	ICRQ PON	26-3561-147
148 ONT/ONU-Peak-Data-Rate-Downstream	Peak downstream data rate for ONT/ONU, in bps 64-bit unsigned integer	ICRQ PON	26-3561-148
149 ONT/ONU-Maximum-Data-Rate-Upstream	Maximum upstream data rate for ONT/ONU, in bps 64-bit unsigned integer	ICRQ PON	26-3561-149
150 ONT/ONU-Assured-Data-Rate-Upstream	Assured upstream data rate for ONT/ONU, in bps 64-bit unsigned integer	ICRQ PON	26-3561-150

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
151 PON-Tree-Maximum-Data-Rate-Upstream	Maximum upstream data rate for the PON tree, in bps 64-bit unsigned integer	ICRQ PON	26-3561-151
152 PON-Tree-Maximum-Data-Rate-Downstream	Maximum downstream data rate for the PON tree, in bps 64-bit unsigned integer	ICRQ PON	26-3561-152
155 Expected-Throughput-Upstream	Expected upstream throughput, which is the net data rate reduced by expected rate loss, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-155
156 Expected-Throughput-Downstream DSL	Expected upstream throughput, which is the net data rate reduced by expected rate loss, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-156
157 Attainable-Expected-Throughput-Upstream	Maximum attainable expected upstream throughput, in Kbps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-157
158 Attainable-Expected-Throughput-Downstream	Maximum attainable expected downstream throughput, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-158

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type L2TP AVP Name	Description	L2TP Message Type Access Line Support	Corresponding DSL Forum VSA
159 Gamma-Data-Rate- Upstream	Actual upstream data rate (net data rate) for the local loop, adjusted down by any throughput capability limitations, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-159
160 Gamma-Data-Rate- Downstream	Actual downstream data rate (net data rate) for the local loop, adjusted down by any throughput capability limitations, in Kbps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-160
161 Attainable-Gamma-Data- Rate-Upstream	Maximum attainable upstream data rate (net data rate) for the local loop, adjusted down by any throughput capability limitations, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-161
162 Attainable-Gamma-Data- Rate-Downstream	Maximum attainable downstream data rate (net data rate) for the local loop, adjusted down by any throughput capability limitations, in bps 64-bit unsigned integer	ICRQ DSL (G.fast)	26-3561-162

Table 20: L2TP AVPs That Provide Subscriber Access Line Information (Continued)

L2TP AVP Type	Description	L2TP Message Type	Corresponding DSL Forum VSA
L2TP AVP Name		Access Line Support	
254 IWF Session	Four-octet field indicating whether or not the internetworking function has been performed for the subscriber's PPPoA over PPPoE session	ICRQ DSL	26-3561-254

Connection Speed Updates on the LAC

You can configure the LAC to notify the LNS when the speed of the subscriber connection changes from the values initially communicated to the LNS by AVP 24 (transmit speed) and AVP 38 (receive speed) in Incoming-Call-Connected (ICCN) messages. When configured to do so, the LAC informs the LNS that it can send these updates by including the Connect Speed Update Enable AVP (98) in the ICRQ message when the L2TP session starts up. The absence of the Connect Speed Update Enable AVP (98) in the ICRQ message indicates that the LAC does not send updates for the life of the session.

When the connection speed changes, the DSLAM notifies the ANCP agent. The ANCP agent then notifies the LAC, and the LAC in turn relays this information to the LNS by sending a Connect-Speed-Update-Notification (CSUN) message that includes the updated speeds in a Connect Speed Update AVP (97) for each session. The LAC collects connection speed updates and sends them in a batch to minimize both the performance overhead on the LAC and the amount of traffic generated as a result of these notifications.

The initial speeds in the ICCN messages and updated speeds in CSUN messages are used by CoS to shape the traffic rate for subscriber access lines.

The presence of the Connect Speed Update Enable AVP (98) in the ICRQ message also informs the LNS that the LAC does respond if it receives a Connect-Speed-Update-Request (CSURQ) message from an LNS.

NOTE: The Junos OS does not currently support the sending of CSURQ messages by MX Series routers configured as an LNS. All discussion about CSURQ messages is strictly about how an MX Series LAC responds to a CSURQ that it receives from a third-party LNS.

A third-party LNS can send a CSURQ message at any time during the life of a tunnel to request the current transmit and receive connection speed for one or more L2TP sessions. The LNS includes the remote (relative to the LNS) session IDs in the CSURQ message. If the LAC has previously sent the Connect Speed Update Enable AVP (98) for the requested sessions, then it responds to the CSURQ with a CSUN message that includes the Connect Speed Update AVP (97) for each session. If no changes to connection speeds have occurred by this time, the LAC simply includes the initial connection speed values that were reported in AVP 24 and AVP 38.

When you enable connect speed updates either globally or for a specific LNS, the LAC does not send CSUN messages unless you have also configured the `tx-connect-speed` statement to be either `ancp` or `service-profile`.

Connection Speed Updates on the LNS

Starting in Junos OS Release 17.4R1, an MX Series router configured as an LNS can process subscriber access line information and connection speed updates that it receives from the LAC. The MX Series router cannot send CSURQ messages to solicit updates from the LAC.

The initial speeds in the ICCN messages and updated speeds in CSUN messages are used by CoS to shape the traffic rate for subscriber access lines.

Interaction Between Global and Per-Destination Configurations

You can configure the LAC to forward the access line information in the ICRQ message that it sends to the LNS and you can configure the LNS to receive and process that information. You can configure this globally for all destinations (endpoints) or for a specific destination. The per-destination configuration enables you to limit transmission to an individual LNS or to a set of LNSs or reception from an individual LAC or a set of LACs. This is useful when you know that some remote gateways do not support this feature or have an incorrect implementation.

Include the `access-line-information` statement at one or both of the following hierarchy levels on the LAC or LNS, respectively, to configure the LAC to forward the access line information in the ICRQ message that it sends to the LNS, or to configure the LNS to receive and process that information:

- `[edit services l2tp]`—Configures forwarding globally for all destinations.
- `[edit services l2tp destination ip-address]`—Configures forwarding for a specific destination.

To configure the LAC to send connection speed updates or the LNS to receive and process the updates, include the `connection-speed-update` option with the `access-line-information` statement at the appropriate hierarchy level on the LAC or LNS, respectively.

The global and per-destination settings interact in the following way:

- Access line information—When forwarding by the LAC or processing by the LNS is enabled globally, you cannot disable the global setting for a specific destination.
- Connection speed updates—When forwarding by the LAC or processing by the LNS is enabled globally, you can disable the global setting for a specific destination (LNS or LAC) by specifying `access-line-information` for the destination and omitting `connection-speed-update`.

Transmission of Tx and Rx Connection Speeds from LAC to LNS

IN THIS SECTION

- [Methods for Determining the Speed Values Reported to the LNS | 233](#)
- [Determining Initial Connect Speeds | 237](#)
- [Fallback Mechanism for Connect Speed Values | 238](#)

An L2TP access concentrator (LAC) uses Incoming-Call-Connected (ICCN) messages during the establishment of an L2TP tunnel session to send attribute-value pairs (AVP) that convey to the L2TP network server (LNS) the subscriber session's connection speed. AVP 24 includes the transmit (Tx) connect speed and AVP 38 includes the receive (Rx) connect speed.

- The L2TP transmit connect speed is the transmit connect speed in bits per second (bps) of the subscriber's access interface; that is, it represents the speed of the connection downstream from the LAC to the subscriber from the perspective of the LAC.
- The L2TP receive connect speed is the speed in bps of the connection upstream from the subscriber to the LAC, again from the perspective of the LAC. When the receive connect speed is different from the transmit connect speed, AVP 38 is included in the ICCN to convey the receive connect speed.

When the connection speed is the same in both directions, the LNS uses the value in AVP 24 for both transmit and receive connect speeds. In this case, the LAC does not send AVP 38. You can override this default behavior by including the `rx-connect-speed-when-equal` statement, which causes the LAC to send AVP 38 even when the transmit and connect speeds are the same. See ["Transmission of the Receive Connect Speed AVP When Transmit and Receive Connect Speeds are Equal "](#) on page 241.

- The Tx and Rx connect speeds sent in the ICCN message are derived from the method determined by the LAC fallback procedure. Because service activation does not occur until after the ICCN is sent, the LAC always falls back to the next method when `service-profile` is configured as the method. When

the service profile is later activated, corresponding speed changes are sent in update messages to the LNS.

- After the L2TP session is established, the Tx and Rx connect speeds can change at any time. When configured to do so, the LAC sends the updated values for each session to the LNS in Connect-Speed-Update-Notification (CSUN) messages. The updated speeds are conveyed in the Connect Speed Update AVP (97).

Methods for Determining the Speed Values Reported to the LNS

The values reported to the LNS can be derived in the following ways:

- You can configure a method globally for the LAC with the `tx-connect-speed-method` statement at the `[edit services l2tp]` hierarchy level. You can specify any of the following methods to determine the source for connect speeds:

NOTE: Starting in Junos OS Release 13.3R1, availability and support for methods vary by Junos OS Release, as described in [Table 21 on page 236](#). The following list includes all historical methods; some of the methods may not be supported in the software release you are using.

- `actual`—The speed is the actual rate of the downstream traffic enforced at the session scheduler node based on local traffic control policy. Only the transmit connect speed is available with this method, so the receive transmit speed is determined by the fallback scheme. Use the `actual` method when you need the reported value to be the downstream speed enforced by the local CoS policy. Other methods may vary from this enforced value.

The `actual` method is supported only when the effective `shaping-rate` statement is included at the `[edit chassis]` hierarchy level. The CLI commit check fails if `actual` is configured but the effective shaping rate is not configured.

No commit check is performed when the Tunnel-Tx-Speed-Method VSA (26-94) is set, so a system log message is generated in this situation to remind the user to configure the effective shaping rate.

- `anccp`—The speed is the adjusted ANCCP-sourced upstream and downstream value that results from a configured percentage correction to the actual ANCCP values. The adjustment is applied on a per-DSL basis to account for ATM encapsulation differences between the BNG and the access-loop and for Layer 1 transport overhead. The initial rate sent to the LNS is the ANCCP value reported at the time the ICCN is sent. Any subsequent changes are sent as updates to the LNS in the CSUN message.

- `none`—This option prevents the LAC from sending either AVP 24 or AVP 38 in the ICCN message; consequently no CSUN messages are sent, either. The LNS has to establish its own upstream and downstream policy in the absence of these values. This option overrides the Juniper Networks RADIUS VSAs, Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163), as well as any other method configured for the connect speed.
- `pppoe-ia-tags`—The speed is derived from the value sent from the DSLAM to the LAC in the Point-to-Point Protocol over Ethernet (PPPoE) intermediate agent (IA) tags. For Ethernet interfaces, the speed is an unadjusted value; for ATM interfaces, the value might be an adjusted value if the tag includes the Encapsulation Overhead attribute (0x90).

This speed value is transmitted when the L2TP session is established. Although the PPPoE IA tag value does not change during a session, the speed reported to the LAC can change. For example, suppose the configured method is `service-profile`. The profile is not activated before the ICCN is sent, and falls back to the PPPoE IA tag, which is sent in the ICCN message. When the service profile is activated later, the service profile rates are sent in an update message (if updates are configured).

- `service-profile`—Depending on your Junos OS release, there are two ways to use service profiles to provide connection speeds. One method uses the speeds from the service profile only in CSUN messages, the other method in ICCN messages.
 - In CSUN messages—The downstream (Tx) speed is derived from the actual CoS that is enforced on the L3 node based on local policy. The upstream (Rx) speed is taken from the configured value in the service profile; no adjustment is made to this value.

By default, service profiles are not activated before the subscriber session is established, so this method falls back to another method for the values sent in the ICCN. When the profile is later activated, then those rates are sent to the LNS in a CSUN message, if updates are enabled.

- In ICCN messages—Starting in Junos OS Release 18.1R1, you can use a dynamic service profile to provide the connection speeds included in AVP 38 and AVP 24 in the ICCN message when the L2TP session is negotiated. At subscriber login, `authd` determines whether the service profile name conveyed in the Juniper Networks Activate-Service VSA (26-65) in the RADIUS Access-Accept message matches the service profile name configured with the `service-rate-limiter` statement at the `[edit access]` hierarchy level. If the names match, the speeds are derived either from default values in the service profile or from parameters passed by the VSA. See ["Specifying a Rate-Limiting Service Profile for L2TP Connection Speeds" on page 253](#) for more information about this method.

The `service-profile` method is supported only when the effective `shaping-rate` statement is included at the `[edit chassis]` hierarchy level. The CLI commit check fails when `service-profile` is configured but the effective shaping rate is not configured.

No commit check is performed when the Tunnel-Tx-Speed-Method VSA (26-94) is set, so a system log message is generated in this situation to remind the user to configure the effective shaping rate.

BEST PRACTICE: We recommend that you use only one service profile per subscriber session to affect the downstream shaping rate or report an upstream rate. If more than one dynamic service profile is applied to the subscriber session such that each affects the downstream shaping rate or reports the upstream rate, the values from the most recently applied profile are reported by L2TP. Deactivation of the most recently applied service does not result in L2TP reporting the upstream speed for an existing (active) service profile.

- **static**—This method causes the LAC to derive the speed from the configured static Layer 2 speed. For Ethernet VLANs, this is the recommended (advisory) shaping rate configured on the PPPoE logical interface underlying the subscriber interface. If the advisory shaping rate is not configured on the underlying interface, then the actual speed of the underlying physical port is used.
- Starting in Junos OS Release 15.1R1, you can configure speed values directly in the Juniper Networks VSAs, Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163). These VSAs may be returned in the RADIUS Access-Accept message. If only one of the VSAs is present, the LAC uses a connect speed method to determine the value for the other speed. To use these VSAs, you must configure RADIUS according to your RADIUS server documentation.
- Starting in Junos OS Release 15.1R1, you can configure a method that is conveyed in the Juniper Networks VSA, Tunnel-Tx-Speed-Method (26-94). If configured, this VSA is returned in the RADIUS Access-Accept message for individual subscribers. The VSA value applies globally rather than to a specific tunnel. The method configured in this VSA specifies the resource that the LAC uses to set the speed. To use this VSA, you must configure RADIUS according to your RADIUS server documentation.
- When the speeds cannot be determined in any other manner, the port speed of the subscriber interface is used.

Table 21 on page 236 lists the available methods by release.

NOTE: Some methods available in VSA 26-94 are not available in the CLI. When one of these methods is received in the VSA, it is translated to a supported method instead of being rejected, or it falls back to another method.

Table 21: Methods for Determining Connect Speeds by Junos OS Release.

Junos OS Release Number	CLI (tx-connect-speed-method)	VSA 26-94 (Tunnel-Tx-Speed-Method)
17.2 and higher	<ul style="list-style-type: none"> • ancp • none • pppoe-ia-tags • service-profile • static (default) 	<ul style="list-style-type: none"> • actual—Translated to service-profile • ancp • CoS—Translated to service-profile • dynamic Layer 2—Translated to static • none • pppoe-ia-tags • service-profile • static
15.1, 16.1, 16.2, 17.1	<ul style="list-style-type: none"> • actual (default) • ancp • none • pppoe-ia-tags 	<ul style="list-style-type: none"> • actual • ancp • CoS—Translated to actual • dynamic Layer 2—Translated to static, which falls back to the port speed of the subscriber access interface • none • pppoe-ia-tags • static—Falls back to the port speed of the subscriber access interface

Table 21: Methods for Determining Connect Speeds by Junos OS Release. (Continued)

Junos OS Release Number	CLI (tx-connect-speed-method)	VSA 26-94 (Tunnel-Tx-Speed-Method)
13.3, 14.1, 14.2	<ul style="list-style-type: none"> • ancp • none • pppoe-ia-tags • static (default) 	n/a

NOTE: Changing the connect speed method in VSA 26-94 or in the CLI configuration has no effect on existing L2TP sessions in which the ICCN has already been sent. All L2TP session negotiations subsequent to the method change use the new setting.

In Junos OS Releases 15.1, 16.1, 16.2, and 17.1 (which support the actual method), the speed values in AVP 24 and AVP 38 are typically not greater than the value that is enforced by CoS on the LAC side of the network. Any difference between the speed reported in these AVPs and that enforced by CoS is attributable to differences between the CoS configuration (of the source that is used to enforce a downstream speed) and the Tx connect speed method used to establish these AVPs.

Determining Initial Connect Speeds

Before the LAC can send initial transmit and receive connect speeds in the ICCN message to the LNS, it has to do the following:

1. Select the method it uses to derive the speeds.
2. Determine the speeds.

The LAC selects the method as follows:

1. If the Tunnel-Tx-Speed-Method VSA (26-94) is present, use the method specified by the VSA value.
2. Otherwise, use the method configured in the CLI with the `tx-connect-speed-method` statement.

The LAC determines the initial speed as follows:

1. If the selected method is `none`, the LAC does not include the transmit and receive speeds in the ICCN.

2. For any other selected method, if the values in the Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163) VSAs are nonzero, the LAC sends those values in the ICCN.
3. If the VSA values are zero, use the selected method determined to derive the values to send.

Consider the following examples:

- VSA 26-94 is received with `ancp` configured as the method. The CLI method is configured as `none`. The LAC selects the VSA 26-94 value, the `ancp` method.

VSA 26-162 and VSA 26-163 are received with nonzero values. The LAC sends these VSA values in the ICCN.

- VSA 26-94 is received with `ancp` configured as the method. The CLI method is configured as `none`. The LAC selects the VSA 26-94 value, the `ancp` method.

VSA 26-162 and VSA 26-163 are received with zero values. The LAC uses the `ancp` method to derive the values to send in the ICCN.

- VSA 26-94 is received with `none` configured as the method. The CLI method is configured as `ancp`. The LAC selects the VSA 26-94 value, `none`, and does not send connect speeds in the ICCN.
- VSA 26-94 is not received. The CLI method is configured as `none`. The LAC does not send connect speeds in the ICCN.

Fallback Mechanism for Connect Speed Values

When the LAC has selected a method to derive the connect speeds, it falls back to a different method in any of the following circumstances:

- One or both connect speed values has not been set by the selected method (VSA 26-94 or the CLI).
- The connect speed value is zero.

When one value is available and nonzero but the other is not, only the unset value falls back to a different method. There is no fallback when the selected method is `none`, because this method prevents the LAC from reporting the connect speeds. The fallback procedure can vary by Junos OS release.

Consider the following examples:

- The selected method is ANCP. The ANCP value for the receive speed is found to be zero. The LAC sends the ANCP value for the transmit speed, but the receive value falls back to the PPPoE IA tag method. The LAC sends the IA tag value for the receive speed.
- The selected method is ANCP. The ANCP value for the receive speed is found to be zero. The LAC sends the ANCP value for the transmit speed, but the receive value falls back to the PPPoE IA tag

method. The IA tag value for the receive speed is also found to be zero, so it falls back to the static Layer 2 method. This is available, so the LAC sends the static Layer 2 value for the receive speed.

- The selected method is service profile. The service profile is not activated before the ICCN is sent, so the LAC falls back to the ANCP method. Both transmit and receive ANCP values are available and nonzero, so the LAC sends these values in the ICCN.

The service profile is activated by a Change of Authorization (CoA) at some later time for the session. If updates are enabled, the LAC sends the service profile values to the LNS in a CSUN message. If updates are not enabled, the service profile values are not reported to the LNS.

Note that updates require the method to be configured in the CLI. Consequently, VSA 26-94 must not be configured or received so that the service profile method is selected from the CLI configuration.

Starting in Junos OS Release 17.2R1, the LAC fallback procedure is as described in [Table 22 on page 239](#).

Table 22: LAC Fallback Procedure When a Connect Speed Value is Not Set (Junos OS Release 17.2 and Higher)

Method	Transmit and Receive Speed Not Set	Transmit Speed Not Set	Receive Speed Not Set
None	No fallback.	No fallback.	No fallback.
Service profile	Both fall back to ANCP method.	Transmit speed falls back to ANCP method.	Receive speed falls back to ANCP method.
ANCP	Both fall back to PPPoE IA tags method.	Transmit speed falls back to PPPoE IA tags method.	Receive speed falls back to PPPoE IA tags method.
PPPoE IA tags	Both fall back to static Layer 2 method.	Transmit speed falls back to static Layer 2 method.	Receive speed falls back to static Layer 2 method.
Static Layer 2	Both fall back to port speed.	Transmit speed falls back to port speed.	Receive speed falls back to transmit speed.

Starting in Junos OS Release 15.1R1, the LAC fallback procedure is as described in [Table 23 on page 240](#).

Table 23: LAC Fallback Procedure When a Connect Speed Value is Not Set (Junos OS Releases 15.1, 16.1, 16.2, 17.1)

Method	Transmit and Receive Speed Not Set	Transmit Speed Not Set	Receive Speed Not Set
None	No fallback.	No fallback.	No fallback.
Actual	Both fall back to ANCP method.	Transmit speed falls back to ANCP method.	Receive speed falls back to ANCP method.
ANCP	Both fall back to PPPoE IA tags method.	If PPPoE IA tags are available for both, then both fall back to PPPoE IA tags method. Otherwise, transmit speed falls back to PPPoE IA tags method.	If PPPoE IA tags are available for both, then both fall back to PPPoE IA tags method. Otherwise, receive speed falls back to PPPoE IA tags method.
PPPoE IA tags	Both fall back to port speed.	Transmit speed falls back to port speed.	Receive speed falls back to port speed.

Starting in Junos OS Release 13.3R1, the LAC fallback procedure is as described in [Table 24 on page 240](#).

Table 24: LAC Fallback Procedure When a Connect Speed Value is Not Set (Junos OS Releases 13.3, 14.1, 14.2)

Method	Transmit and Receive Speed Not Set	Transmit Speed Not Set	Receive Speed Not Set
None	No fallback.	No fallback.	No fallback.
ANCP	Both fall back to PPPoE IA tags method.	If PPPoE IA tags are available for both, then both fall back to PPPoE IA tags method. Otherwise, transmit speed falls back to PPPoE IA tags method.	If PPPoE IA tags are available for both, then both fall back to PPPoE IA tags method. Otherwise, receive speed falls back to PPPoE IA tags method.
PPPoE IA tags	Both fall back to static Layer 2 method.	Transmit speed falls back to static Layer 2 method.	Receive speed falls back to static Layer 2 method.

Table 24: LAC Fallback Procedure When a Connect Speed Value is Not Set (Junos OS Releases 13.3, 14.1, 14.2) (Continued)

Method	Transmit and Receive Speed Not Set	Transmit Speed Not Set	Receive Speed Not Set
Static Layer 2	Both fall back to port speed.	Transmit speed falls back to port speed.	Receive speed falls back to transmit speed.

NOTE: For both Gigabit Ethernet (ge) and 10-Gigabit Ethernet (xe) interfaces, the port speed value is set to 1,000,000,000. For aggregated Ethernet (ae) interfaces, the port speed value is set to 0. The port speed value for all these interface types is reported in both AVP 24 and AVP 38.

Transmission of the Receive Connect Speed AVP When Transmit and Receive Connect Speeds are Equal

The L2TP Rx Connect Speed (in bits per second) AVP, which is represented by AVP 38, is included in the ICCN message when the receive connect speed is different from the transmit connect speed. By default, when the connection speed is the same in both directions, AVP 38 is not sent; the LNS uses the value in AVP 24 for both transmit and receive connect speeds.

AVP 38 is generated when the receive connect speed of the access interface is set equal to the calculated transmit connect speed by issuing the `rx-connect-speed-when-equal` statement at the `[edit services l2tp]` hierarchy level. In this scenario, the LAC transmits the same value for transmit and receive connect speeds that are sent to the LNS through the AVP 24 and AVP 38 in the ICCN message.

To configure the sending of AVP 38 when the connection speeds are the same in both the downstream and upstream directions:

- Configure the transmission of the receive connect speed, AVP 38, when the receive connect speed is set equal to the calculated transmit connect speed.

```
[edit services l2tp]
user@host# set rx-connect-speed-when-equal
```

Configuring the Method to Derive the LAC Connection Speeds Sent to the LNS

The LAC connection speeds are determined in one of several ways:

- The Juniper Networks VSAs, Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163).
- The Juniper Networks VSA, Tunnel-Tx-Speed-Method (26-94).
- The CLI configuration.
- The port speed of the subscriber access interface.

You can include the `tx-connect-speed-method` statement at the `[edit services l2tp]` hierarchy level to configure a method that specifies the resource that the LAC uses for setting these speeds when the Juniper Networks VSAs are not returned for the subscriber.

Starting in Junos OS Release 17.2R1, when you enable connect speed updates for the LAC you must include the `tx-connect-speed-method` statement. You also must specify either `ancp` or `service-profile` as the method; otherwise, the LAC does not send CSUN messages.

Changing the connect speed method in the CLI configuration or in VSA 26-94 has no effect on existing L2TP sessions in which the ICCN has already been sent. All L2TP session negotiations subsequent to the method change use the new setting.

NOTE: Starting in Junos OS Release 13.3R1, availability and support for methods vary by Junos OS Release. The following procedure lists all historical methods; some of the methods may not be supported in the software release you are using. See "[Transmission of Tx and Rx Connection Speeds from LAC to LNS](#)" on page 232 for a table of support by release.

To set the method for calculating the transmit connect speed:

- (Optional) Configure the LAC to use the class-of-service effective shaping rates.

```
[edit services l2tp]
user@host# set tx-connect-speed-method actual
```

NOTE: This method requires that the effective shaping rate statement is configured at the `[edit chassis]` hierarchy level. If it is not, then committing this method fails. However, if the method

is received from RADIUS in VSA 26-94, a system log message is generated instead, because no commit check is performed in this case.

- (Optional) Configure the LAC to use the values derived from the ANCP value configured on the PPPoE interface underlying the subscriber interface.

```
[edit services l2tp]
user@host# set tx-connect-speed-method ancp
```

- (Optional) Configure the LAC to use the values provided in the PPPoE IA tags received from the DSLAM.

```
[edit services l2tp]
user@host# set tx-connect-speed-method pppoe-ia-tags
```

In this case, the value of Actual-Data-Rate-Downstream (VSA 26-129) is used for AVP 24. The value of Actual-Data-Rate-Upstream (VSA 26-130) is used for AVP 38 and is sent only when the VSA values differ.

NOTE: This speed derived from the IA tags does not apply to subscribers that are already logged in; it is effective only for subscribers that log in after this setting has been saved.

- (Optional) Configure the LAC to use the following:
 - Downstream (Tx) speed: The actual CoS rate that is enforced on the level 3 node based on local policy
 - Upstream (Rx) speed: The value configured in the dynamic service profile.

1. Specify the `service-profile` method.

```
[edit services l2tp]
user@host# set tx-connect-speed-method service-profile
```

2. In the dynamic service profile, configure the ingress shaping rate from CoS to be used by the LAC to report to the LNS as the Rx connect speed.

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number]
user@host# set report-ingress-shaping-rate bps
```

NOTE: The service-profile method requires that the effective shaping rate statement is configured at the [edit chassis] hierarchy level. If it is not, the commit check fails. However, if the service-profile method is received from RADIUS in VSA 26-94, a system log message is generated instead, because no commit check is performed in this case.

NOTE: For another method to use service profiles to provide the connection speeds, see ["Specifying a Rate-Limiting Service Profile for L2TP Connection Speeds" on page 253](#).

- (Optional) Configure the LAC to use the underlying interface's recommended (advisory) downstream shaping rate for AVP 24 and recommended upstream shaping rate for AVP 38. This is also referred to as the static Layer 2 shaping rate.

```
[edit services l2tp]
user@host# set tx-connect-speed-method static
```

You configure the advisory rates under the PPPoE logical interface underlying the subscriber interface with the advisory-options statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level. If the advisory speed is not configured, then the actual port speed is used. For ge and xe interfaces, the speed value is set to 10,000,000 and for ae interfaces, the speed value is set to 0 and sent in both AVP 24 and AVP 38

- (Optional) Configure the LAC to disable sending AVP 24 and AVP 38.

NOTE: This option prevents the LAC from sending either AVP 24 or AVP 38 in the ICCN messages. This option also overrides the Juniper Networks RADIUS VSAs, Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163).

```
[edit services l2tp]
user@host# set tx-connect-speed-method none
```

Configuring the Reporting and Processing of Subscriber Access Line Information

The L2TP AVP extensions defined in RFC 5515, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extension*, enable the LAC to report to the LNS characteristics of the subscriber's access line, such as identification attributes, line type, connection speed, various data rates, and so on. The LAC receives the access line information when the subscriber's CPE initiates a connection request, and forwards the available information in various AVPs included in ICRQ messages to the LNS. The LAC can also signal to the LNS that it is capable of sending updates to the subscriber connection speeds; these are conveyed by the Connect Speed Update AVP (97) in the CSUN message.

Starting in Junos OS Release 17.4R1, RFC 5515 AVP extensions are also supported on the LNS. Consequently, you can configure the LNS to process subscriber access line information and connection speed updates that it receives from the LAC.

Starting in Junos OS Release 19.3R1, AVPs for PON and G.fast access lines are supported, corresponding to the Broadband Forum PON and G.fast TLVs.

NOTE: Subscriber access line information conveyed by AVPs in ICRQ messages is passed to RADIUS in DSL Forum VSA AVPs. Initial and updated connection speeds conveyed in ICCN and CSUN messages can be used by CoS to adjust traffic rates for the subscriber lines.

By default, neither the access line information forwarding or connection speed update capability are enabled on the LAC. You must configure the capabilities for all LNS endpoints or for a specific LNS endpoint. The per-destination configuration applies to all tunnels with that destination IP address. You might want to use a per-destination configuration when you know that only certain endpoints support or correctly implement this feature.

Similarly, processing of this information by the LNS is not enabled by default. You can enable processing for information received from all LAC endpoints or for specific LAC endpoints. The per-destination configuration applies to all tunnels with that destination IP address.

NOTE: The CLI statements are the same for both the LAC and LNS; the difference is that you include the statements in the LAC configuration or the LNS configuration.

To configure the LAC to send information about subscriber access lines to the LNS, or to configure the LNS to process this information received from the LAC:

- Configure the capability globally for all endpoints.

```
[edit services l2tp]
user@host# set access-line-information
```

- Configure the capability for a specific endpoint.

```
[edit services l2tp destination address ip-address]
user@host# set access-line-information
```

BEST PRACTICE: Do not configure the `connection-speed-update` option on the LAC when the LNS does not support connection speed changes. This might be an LNS that is not configured to process the updates or a noncompliant, third-party LNS. Configuring the LAC option for such an LNS generates additional control messages that are ignored.

To configure the LAC to also send updates to the LNS about changes in connection speed, or to configure the LNS to process speed updates received from the LAC:

- Include the update option when you configure the capability.

```
[edit services l2tp]
user@host# set access-line-information connection-speed-update
```

or

```
[edit services l2tp destination address ip-address]
user@host# set access-line-information connection-speed-update
```

- When you configure the LAC to send updates, you must also configure the method by which the connect speed values are derived. The method specifies the source of the update values. On the LNS, the derivation method is not relevant and cannot be configured.

```
[edit services l2tp]
user@host# set tx-connect-speed-method method
```

Consider the following examples:

- The following configuration specifies that for all tunnels with an endpoint address of 192.0.2.2, the LAC reports access line characteristics sourced from the ANCP agent or the PPPoE intermediate agent (in that order) to the LNS in the ICRQ message. The Connect Speed Update Enable AVP (98) is not included in the ICRQ; consequently no CSUN messages are sent to the LNS to report speed changes in the subscriber access lines reported by the ANCP agent. The LAC ignores any CSURQ messages that it receives from the LNS; this can be only a third-party LNS, because the sending of CSURQ messages is not supported on MX Series routers configured as an LNS.

```
[edit services l2tp destination address 192.0.2.2]
user@host# set access-line-information
```

- The following configuration specifies that for all tunnels with an endpoint address of 203.0.113.23, the LAC reports access line characteristics sourced from the ANCP agent or the PPPoE intermediate agent (in that order) to the LNS in the ICRQ message. The Connect Speed Update Enable AVP (98) is included in the ICRQ; CSUN messages are sent to the LNS to report speed changes in the subscriber access lines reported by the ANCP agent. The LAC accepts any CSURQ messages that it receives from the LNS and responds with a CSUN message; this can be only a third-party LNS, because the sending of CSURQ messages is not supported on MX Series routers configured as an LNS.

```
[edit services l2tp]
user@host# set destination address 203.0.113.23 access-line-information connection-speed-update
user@host# set tx-connect-speed-method ancip
```

When access line information forwarding is enabled globally, you cannot disable it for a specific destination. However, when connection speed updates are enabled globally, you can disable updates for a specific destination.

- The following configuration specifies that both forwarding of access line characteristics and connection speed updates are enabled for all destinations. For destination 198.51.100.2, the global updates configuration is overridden by repeating the access line configuration for, and omitting the connection speed updates for, that destination.

```
[edit services l2tp]
user@host# set access-line-information connection-speed-update
user@host# set tx-connect-speed-method ancp
[edit services l2tp destination address 198.51.100.2]
user@host# set access-line-information
```

The `show services l2tp summary` command displays the configuration that applies to all destinations. The following sample output confirms the global configuration in this example:

```
user@host> show services l2tp summary
Failover within a preference level is Disabled
  Weighted load balancing is Disabled
  Tunnel authentication challenge is Enabled
  Calling number avp is Enabled
  Failover Protocol is Disabled
  Tx Connect speed method is static
  Rx speed avp when equal is enabled
  Tunnel assignment id format is assignment-id
  Tunnel Tx Address Change is Accept
  Min Retransmissions Timeout for control packets is 2 seconds
  Max Retransmissions for Established Tunnel is 7
  Max Retransmissions for Not Established Tunnel is 5
  Tunnel Idle Timeout is 60 seconds
  Destruct Timeout is 300 seconds
  Destination Lockout Timeout is 300 seconds
  Access Line Information is Enabled, Speed Updates is Enabled
Destinations: 0, Tunnels: 0, Sessions: 0, Switched sessions: 0
```

The `show services l2tp destination detail` command displays the configuration for each destination individually. The following sample output verifies that connection speed updates are disabled for 198.51.100.2:

```
user@host> show services l2tp destination detail
Local name: 1
  Remote IP: 198.51.100.2
  Tunnels: 1, Sessions: 1
  State: Enabled
  Local IP: 203.0.113.2
  Transport: ipUdp, Logical System: default, Router Instance: default
  Lockout State: not locked
  Access Line Information: Enabled, Speed Updates: Disabled
...
```

- In this example, the forwarding of access line characteristics is enabled for all destinations, but connection speed updates are enabled for only one destination, 198.51.100.21.

```
[edit services l2tp]
user@host# set access-line-information
[edit services l2tp destination address 198.51.100.21]
user@host# set access-line-information connection-speed-update
user@host# up
user@host# set tx-connect-speed-method ancp
```

The following sample output confirms that connection speed updates are disabled globally:

```
user@host> show services l2tp summary
Failover within a preference level is Disabled
  Weighted load balancing is Disabled
  Tunnel authentication challenge is Enabled
  Calling number avp is Enabled
  Failover Protocol is Disabled
  Tx Connect speed method is static
  Rx speed avp when equal is enabled
  Tunnel assignment id format is assignment-id
  Tunnel Tx Address Change is Accept
  Min Retransmissions Timeout for control packets is 2 seconds
  Max Retransmissions for Established Tunnel is 7
  Max Retransmissions for Not Established Tunnel is 5
```

```
Tunnel Idle Timeout is 60 seconds
Destruct Timeout is 300 seconds
Destination Lockout Timeout is 300 seconds
Access Line Information is Enabled, Speed Updates is Disabled
Destinations: 0, Tunnels: 0, Sessions: 0, Switched sessions: 0
```

The following sample output confirms that connection speed updates are enabled for destination 198.51.100.21:

```
user@host> show services l2tp destination detail
Local name: 1
  Remote IP: 198.51.100.21
  Tunnels: 1, Sessions: 1
  State: Enabled
  Local IP: 203.0.113.3
  Transport: ipUdp, Logical System: default, Router Instance: default
  Lockout State: not locked
  Access Line Information: Enabled, Speed Updates: Enabled
...
```

Preventing the LAC from Sending Calling Number AVP 22 to the LNS

Calling Number AVP 22 typically identifies the interface that is connected to the customer in the access network. When RADIUS includes the Calling-Station-Id in the Access-Accept message, that value is used for the Calling Number AVP. Otherwise, the underlying interface (for example, the S-VLAN IFL) on which the PPPoE session is established is used for the Calling Number AVP value.

By default, the LAC includes this AVP in the incoming-call request (ICRQ) packets that it sends to the LNS. However, you may wish to hide your network access interface information. To do so, you can configure the tunnel so that the LAC does not send the Calling Number AVP to the LNS.

To disable sending the Calling Number AVP:

- Configure disabling.

```
[edit services l2tp]
user@host# set disable-calling-number-avp
```

Override the Calling-Station-ID Format for the Calling Number AVP

The LAC sends information about the access line or the subscriber to the LNS in L2TP Calling Number AVP 22. This AVP is conveyed in the incoming-call request (ICRQ) packet when the L2TP session is being established. AVP 22 by default identifies the access node interface that is connected to the customer in the access network; this is the agent circuit identifier or ACI. The LAC receives the ACI in the PPPoE Active Discovery Request (PADR) packet from the L2TP client as DSL Forum Agent-Circuit-ID VSA [26-1].

Alternatively, you can use the `calling-station-id-format` statement to change the values sent in the AVP. For example, you might specify that the agent remote identifier (ARI) received in the PADR as DSL Forum Agent-Remote-ID VSA [26-2] is used instead of the agent circuit identifier, that both are used, or that additional attributes are included. The set of values used in the AVP is known as the Calling-Station-ID format. When this is configured, then the value of the AVP is subsequently sent to the RADIUS server as Calling-Station-ID attribute (31). See *Configuring a Calling-Station-ID with Additional Options* for more information.

In some cases you may want the value of Calling Number AVP 22 to be independent from the RADIUS attribute value. You can do this by overriding the configured Calling-Station-ID format for the value. Use the `remote-circuit-id-format` statement to specify a different format for the AVP: the ACI, the ARI, or both the ACI and ARI from the PADR packet.

You can also configure fallback values that are sent in the Calling Number AVP when the values you configure with the `remote-circuit-id-format` statement are not present in the PADR. You can configure the fallback option to send the configured Calling-Station-ID or the default underlying interface as the calling number AVP.

Before you begin:

- Configure an access profile.
- Configure L2TP.
- Configure RADIUS.

To configure the override in the access profile:

1. Configure the LAC to send the calling number AVP using the configured remote circuit ID format instead of the Calling-Station-ID format.

NOTE: The override statement fails commit check if you have not configured the `remote-circuit-id-format` statement.

```
user@host# set access profile profile-name override calling-station-id remote-circuit-id
```

2. Configure the format of the values that override the Calling-Station-ID in AVP 22. You can configure the format to include the ACI, the ARI, or both the ACI and ARI.

```
user@host# set access profile profile-name radius options remote-circuit-id-format agent-circuit-id
user@host# set access profile profile-name radius options remote-circuit-id-format agent-remote-id
```

[Table 25 on page 252](#) describes the attributes sent in calling number AVP 22 based on the attributes received in the PADR and the format configured in the `remote-circuit-id-format` configuration statement.

Table 25: Attributes Sent as Calling Number AVP Based on Remote Circuit ID Format and Attributes Received in PADR

Remote Circuit ID Format	Attributes Received in PADR	Attributes Sent in Calling Number AVP
Agent-Circuit-ID	Agent-Circuit-ID, Agent-Remote-ID	Agent-Circuit-ID
Agent-Remote-ID	Agent-Circuit-ID, Agent-Remote-ID	Agent-Remote-ID
Agent-Circuit-ID, Agent-Remote-ID	Agent-Circuit-ID, Agent-Remote-ID	Agent-Circuit-ID, Agent-Remote-ID
Agent-Circuit-ID, Agent-Remote-ID	Agent-Circuit-ID	Agent-Circuit-ID
Agent-Circuit-ID, Agent-Remote-ID	Agent-Remote-ID	Agent-Remote-ID

3. (Optional) Configure the fallback value to be used. Fallback is triggered if the ACI and ARI are not present in the PADR but are configured in the remote circuit ID format. You can configure the LAC to send the configured Calling-Station-ID or the default underlying interface in the Calling number AVP 22 when fallback is triggered.

```
user@host# set access profile profile-name remote-circuit-id-fallback configured-calling-
station-id
user@host# set access profile profile-name remote-circuit-id-fallback default
```

The remote circuit ID format determines what triggers the fallback. [Table 26 on page 253](#) shows the fallback trigger based on the remote circuit ID format.

Table 26: Fallback Trigger for Remote Circuit ID Format

Remote Circuit ID Format	Fallback Trigger
Agent-Circuit-ID	Agent-Circuit-ID is empty
Agent-Remote-ID	Agent-Remote-ID is empty
Agent-Circuit-ID, Agent-Remote-ID	Both Agent-Circuit-ID and Agent-Remote-ID are empty

4. (Optional) Configure an alternative delimiter character that the router uses to separate the concatenated values in the resulting remote circuit ID string when more than one value is specified in the remote circuit ID format. The default delimiter is a hash character (#).

```
user@host#set access profile profile-name remote-circuit-id-delimiter "delimiter"
```

Specifying a Rate-Limiting Service Profile for L2TP Connection Speeds

When an L2TP session is negotiated, the LAC sends to the LNS an ICCN message that includes values for the Rx connection speed (in AVP 38) and Tx connection speed (in AVP 24) at the LAC. The LAC uses values from the best source available at the time of negotiation. If multiple sources are available, the selection is made based on preference hierarchy of the sources. The source is either RADIUS, ANCP, or PPPoE-IA tags.

By default, the LAC cannot use a service profile received in a RADIUS Access-Accept message as the source, because the profile is not applied until the network family is activated, which occurs after the session negotiation completes. However, if the LNS supports *RFC 5515, Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*, the LAC can send a connection speed update to the LNS with values from the service profile.

Starting in Junos OS Release 18.1R1, you can use a dynamic service profile to provide the connection speeds included in AVP 38 and AVP 24 when the L2TP session is negotiated. At subscriber login, `authd` determines whether the configured service profile name matches the profile name conveyed in the Juniper Networks Activate-Service VSA (26-65) in the RADIUS Access-Accept message. If the names match, the speeds are derived either from default values in the service profile or from parameters passed by the VSA.

This processing by `authd` to establish the connection speeds takes place only at subscriber login. It does not occur in response to reauthentication or CoA requests.

NOTE: For this feature to work, you must also use the `tx-connect-speed-method` statement at the `[edit services l2tp]` hierarchy level to set the method to `service-profile`. You must also configure the `effective-shaping-rate` statement at the `[edit chassis]` hierarchy level.

You can define the rates directly in the service profile as default values for user-defined variables. Alternatively, you can configure the rates to be passed by RADIUS in VSA 26-65. In either case, the first value is taken as the receive speed (the upstream rate from the subscriber to the LAC) and the second value is taken as the transmit speed (the downstream rate from the LAC to the subscriber). The VSA might be configured to pass more than two parameters, but only the first two parameters matter for the service rate-limiting function.

The rate values are specified in the profile or VSA 26-65 in Kbps, but the L2TP AVP format requires rate values in bps. When you enable this feature, default multipliers automatically convert the rates from Kbps to bps. You can also configure the multiplier options to adjust the rates up or down. The adjusted values are equivalent to the Juniper Networks RADIUS VSAs, Rx-Connect-Speed (26-163) and Tx-Connect-Speed (26-162). These values are stored as such in the session database. Because the values are available in the SDB before the L2TP connection is negotiated, the LAC includes them in the ICCN message as AVP 38 and AVP 24. They are treated as RADIUS-sourced values and consequently have the highest precedence.

NOTE: A parameter value of zero signifies that the rate is not set. For example, if VSA 26-65 returns `service-profile-name(0, 0)`, then no value is set in the SDB for Rx or Tx.

Another circumstance that causes no values to be set in the SDB is if VSA 26-65 does not pass any parameters and you failed to set default values in the service profile. In this case, there are no values for authd to derive and so nothing to place in the SDB for Rx or Tx.

If the service used to establish the rate limiters is deactivated or deleted, authd then clears those rate limiter values from the subscriber session. If the service is reactivated, authd does not reinstate the rate limiters.

To configure LAC connection speeds to be derived at login from a dynamic service profile and to optionally adjust the rates:

1. Specify the dynamic service profile that supplies the connection speeds.

```
[edit access]
user@host# set service-rate-limiter service-name service-profile-name
```

2. (Optional) Configure a value that is multiplied with the Rx connect speed specified in the service profile.

```
[edit access]
user@host# set service-rate-limiter rx-multiplier rx-multiplier
```

3. (Optional) Configure a value that is multiplied with the Tx connect speed specified in the service profile.

```
[edit access]
user@host# set service-rate-limiter tx-multiplier tx-multiplier
```

4. Set the method for determining the connection speed.

```
[edit services l2tp]
user@host# set tx-connect-speed-method service-profile
```

5. Enable the reporting of the actual downstream rate in RADIUS accounting messages.

```
[edit chassis]
user@host# set effective-shaping-rate
```

For example, suppose you configure a dynamic service policy, l2tp-service. The policy includes user-defined variables, upstream and downstream, with default values, respectively, of 20,000 Kbps and 30,000 Kbps. The upstream variable is used for the input (ingress) filter and downstream variable is used for the output (egress) filter.

```
[edit dynamic-profiles l2tp-service]
user@host# set variables upstream default-value 20000
user@host# set variables downstream default-value 30000
user@host# set variables aggregate default-value 50000
user@host# interfaces pp0 "$junos-interface-unit" family inet filter input "$upstream"
user@host# interfaces pp0 "$junos-interface-unit" family inet filter output "$downstream"
```

Then you configure the following service rate limiter, which specifies that when a service policy named l2tp-service is returned, the Rx value in the policy, or passed by the VSA, is multiplied by 1005. The Tx value is multiplied by 1003.

```
[edit access]
user@host# set service-rate-limiter service-name l2tp-service
user@host# set service-rate-limiter rx-multiplier 1005
user@host# set service-rate-limiter tx-multiplier 1003
```

Suppose a subscriber logs in and the Access-Accept message from the RADIUS server includes the Activate-Service VSA, 26-55, specifying l2tp-service. What happens next depends on the parameters passed by the VSA.

- The VSA includes "l2tp-service" with no parameters. The following values are stored in the SDB:
 - Rx is the default value in the policy multiplied by the configured multiplier:
20000 Kbps x 1005 = 20,100,000 bps.
 - Tx is the default value in the policy multiplied by the configured multiplier:
30000 Kbps x 1003 = 30,090,000 bps.
- The VSA includes "l2tp-service(10000, 15000)". The following values are stored in the SDB:
 - Rx is the first parameter passed by the VSA multiplied by the configured multiplier:
10000 Kbps x 1005 = 10,050,000 bps.
 - Tx is the second parameter passed by the VSA multiplied by the configured multiplier:
15000 Kbps x 1003 = 15,045,000 bps.
- The VSA includes "l2tp-service(10000)". The following values are stored in the SDB:
 - Rx is the first (and only) parameter passed by the VSA multiplied by the configured multiplier:

10000 Kbps x 1005 = 10,050,000 bps.

- Because the VSA does not pass a second parameter, Tx is the default value in the policy multiplied by the configured multiplier:
30000 Kbps x 1003 = 30,090,000 bps.
- The VSA includes "l2tp-service(10000, 0)". The following values are stored in the SDB:
 - Rx is the first parameter passed by the VSA multiplied by the configured multiplier:
10000 Kbps x 1005 = 10,050,000 bps.
 - Because the second parameter passed is zero, and zero means that the rate is not set, no value is stored in the SDB for Tx.
- The VSA includes "l2tp-service(0, 0)". The following values are stored in the SDB:
 - Because a passed value of zero means that the rate is not set, no value is stored in the SDB for either Rx or Tx.
- The VSA includes "l2tp-service(10000, 15000, 4000000)". The following values are stored in the SDB:
 - Rx is the first parameter passed by the VSA multiplied by the configured multiplier:
10000 Kbps x 1005 = 10,050,000 bps.
 - Tx is the second parameter passed by the VSA multiplied by the configured multiplier:
15000 Kbps x 1003 = 15,045,000 bps.

Release History Table

Release	Description
19.3R1	Starting in Junos OS Release 19.3R1, AVPs for PON and G.fast access lines are supported, corresponding to the Broadband Forum PON and G.fast TLVs.
18.1R1	Starting in Junos OS Release 18.1R1, you can use a dynamic service profile to provide the connection speeds included in AVP 38 and AVP 24 in the ICCN message when the L2TP session is negotiated.
17.4R1	Starting in Junos OS Release 17.4R1, an MX Series router configured as an LNS can process subscriber access line information and connection speed updates that it receives from the LAC.
17.4R1	Starting in Junos OS Release 17.4R1, RFC 5515 AVP extensions are also supported on the LNS.
17.2R1	Starting in Junos OS Release 17.2R1, the LAC fallback procedure is as described in Table 22 on page 239 .

17.2R1	Starting in Junos OS Release 15.1R1, the LAC fallback procedure is as described in Table 23 on page 240 .
17.2R1	Starting in Junos OS Release 13.3R1, the LAC fallback procedure is as described in Table 24 on page 240 .
17.2R1	Starting in Junos OS Release 17.2R1, when you enable connect speed updates for the LAC you must include the tx-connect-speed-method statement.
15.1R1	Starting in Junos OS Release 15.1R1, you can configure speed values directly in the Juniper Networks VSAs, Tx-Connect-Speed (26-162) and Rx-Connect-Speed (26-163).
15.1R1	Starting in Junos OS Release 15.1R1, you can configure a method that is conveyed in the Juniper Networks VSA, Tunnel-Tx-Speed-Method (26-94).
14.1	Starting in Junos OS Release 14.1, L2TP supports a set of AVPs that convey information about subscriber access lines from the LAC to the LNS.
13.3R1	Starting in Junos OS Release 13.3R1, availability and support for methods vary by Junos OS Release, as described in Table 21 on page 236 .
13.3R1	Starting in Junos OS Release 13.3R1, availability and support for methods vary by Junos OS Release.

RELATED DOCUMENTATION

[L2TP for Subscriber Access Overview | 144](#)

[Configuring an L2TP LAC | 176](#)

[LAC Tunnel Selection Overview | 183](#)

DSL Forum Vendor-Specific Attributes

Juniper Networks VSAs Supported by the AAA Service Framework

RADIUS Servers and Parameters for Subscriber Access

Filtering RADIUS Attributes and VSAs from RADIUS Messages

L2TP LNS Inline Service Interfaces

IN THIS SECTION

- [Configuring an L2TP LNS with Inline Service Interfaces | 259](#)
- [Applying PPP Attributes to L2TP LNS Subscribers per Inline Service Interface | 261](#)
- [Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile | 264](#)
- [Configuring an L2TP Access Profile on the LNS | 266](#)
- [Configuring a AAA Local Access Profile on the LNS | 268](#)
- [Configuring an Address-Assignment Pool for L2TP LNS with Inline Services | 269](#)
- [Configuring the L2TP LNS Peer Interface | 271](#)
- [Enabling Inline Service Interfaces | 272](#)
- [Configuring an Inline Service Interface for L2TP LNS | 274](#)
- [Configuring Options for the LNS Inline Services Logical Interface | 275](#)
- [LNS 1:1 Stateful Redundancy Overview | 276](#)
- [Configuring 1:1 LNS Stateful Redundancy on Aggregated Inline Service Interfaces | 276](#)
- [Verifying LNS Aggregated Inline Service Interface 1:1 Redundancy | 279](#)
- [L2TP Session Limits and Load Balancing for Service Interfaces | 282](#)
- [Example: Configuring an L2TP LNS | 286](#)
- [Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces | 301](#)
- [Applying Services to an L2TP Session Without Using RADIUS | 304](#)
- [Configuring a Pool of Inline Services Interfaces for Dynamic LNS Sessions | 313](#)
- [Configuring a Dynamic Profile for Dynamic LNS Sessions | 314](#)

Configuring an L2TP LNS with Inline Service Interfaces

The L2TP LNS feature license must be installed before you begin the configuration. Otherwise, a warning message is displayed when the configuration is committed.

To configure an L2TP LNS with inline service interfaces:

1. (Optional) Configure a user group profile that defines the PPP configuration for tunnel subscribers.
See ["Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile" on page 264.](#)

2. (Optional) Configure PPP attributes for subscribers on inline service interfaces.
See ["Applying PPP Attributes to L2TP LNS Subscribers per Inline Service Interface" on page 261.](#)
3. Configure inline IP reassembly.
See ["Configuring IP Inline Reassembly for L2TP" on page 320.](#)
4. Configure an L2TP access profile that defines the L2TP parameters for each LNS client (LAC).
See ["Configuring an L2TP Access Profile on the LNS" on page 266.](#)
5. (Optional) Configure a AAA access profile to override the access profile configured under the routing instance.
See ["Configuring a AAA Local Access Profile on the LNS" on page 268.](#)
6. Configure a pool of addresses to be dynamically assigned to tunneled PPP subscribers.
See ["Configuring an Address-Assignment Pool for L2TP LNS with Inline Services" on page 269.](#)
7. Configure the peer interface to terminate the tunnel and the PPP server-side IPCP address.
See ["Configuring the L2TP LNS Peer Interface" on page 271.](#)
8. Enable inline service interfaces on an MPC.
See ["Enabling Inline Service Interfaces" on page 272.](#)
9. Configure a service interface.
See ["Configuring an Inline Service Interface for L2TP LNS" on page 274.](#)
10. Configure options for each inline service logical interface.
See ["Configuring Options for the LNS Inline Services Logical Interface" on page 275.](#)
11. (Optional) Configure an aggregated inline service interface and 1:1 stateful redundancy.
See ["Configuring 1:1 LNS Stateful Redundancy on Aggregated Inline Service Interfaces" on page 276](#)
12. Configure the L2TP tunnel group.
See ["Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces" on page 301.](#)
13. (Optional) Configure a dynamic profile that dynamically creates L2TP logical interfaces.
See ["Configuring a Dynamic Profile for Dynamic LNS Sessions" on page 314.](#)
14. (Optional) Configure a service interface pool for dynamic LNS sessions.
See ["Configuring a Pool of Inline Services Interfaces for Dynamic LNS Sessions" on page 313.](#)
15. (Optional) Specify how many times L2TP retransmits unacknowledged control messages.
See ["Configuring Retransmission Attributes for L2TP Control Messages" on page 152.](#)
16. (Optional) Specify how long a tunnel can remain idle before being torn down.
See ["Setting the L2TP Tunnel Idle Timeout" on page 171.](#)
17. (Optional) Specify the L2TP receive window size for the L2TP tunnel. The receive window size specifies the number of packets a peer can send before waiting for an acknowledgment from the router.
See ["Setting the L2TP Receive Window Size" on page 171.](#)

18. (Optional) Specify how long the L2TP retains information about terminated dynamic tunnels, sessions, and destinations.
See ["Setting the L2TP Destruct Timeout" on page 172.](#)
19. (Optional) Configure the L2TP destination lockout timeout.
See ["Configuring the L2TP Destination Lockout Timeout" on page 172.](#)
20. (Optional) Configure L2TP tunnel switching.
See ["Configuring L2TP Tunnel Switching" on page 168.](#)
21. (Optional) Prevent the creation of new sessions, destinations, or tunnels for L2TP.
See ["Configuring L2TP Drain" on page 174.](#)
22. (Optional) Configure whether the L2TP failover protocol is negotiated or the silent failover method is used for resynchronization.
See ["Configuring the L2TP Peer Resynchronization Method" on page 324.](#)
23. (Optional) Enable SNMP statistics counters.
See ["Enabling Tunnel and Global Counters for SNMP Statistics Collection" on page 154.](#)
24. (Optional) Configure trace options for troubleshooting the configuration.
See ["Tracing L2TP Events for Troubleshooting" on page 326.](#)

You also need to configure CoS for LNS sessions. For more information, see [Configuring Dynamic CoS for an L2TP LNS Inline Service.](#)

Applying PPP Attributes to L2TP LNS Subscribers per Inline Service Interface

You can configure PPP attributes that are applied by the LNS on the inline service (si) interface to the PPP subscribers tunneled from the LAC. Because you are configuring the attributes per interface rather than with a user group profile, the attributes for subscribers can be varied with a finer granularity. This configuration matches that used for terminated PPPoE subscribers.

To configure the PPP attributes for dynamically created si interfaces:

1. Specify the predefined dynamic interface and logical interface variables in the dynamic profile.

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"
```

2. Configure the interval between PPP keepalive messages for the L2TP tunnel terminating on the LNS.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set keepalives interval seconds
```

3. Configure PPP authentication methods that apply to tunneled PPP subscribers at the LNS.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set ppp-options chap
user@host# set ppp-options pap
```

4. Specify a set of AAA options that is used for authentication and authorization of tunneled PPP subscribers at the LNS that are logging in by means of the subscriber and AAA contexts that are specified in the AAA options set.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set ppp-options aaa-options aaa-options-name
```

The option set is configured with the `aaa-options aaa-options-name` statement at the `[edit access]` hierarchy level.

5. Configure the router to prompt Customer Premises Equipment (CPE) to negotiate both primary and secondary DNS addresses during IPCP negotiation for tunneled PPP subscribers at the LNS.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set ppp-options ipcp-suggest-dns-option
```

6. (Optional) Disable validation of the PPP magic number during LCP negotiation and in LCP keepalive (echo-request/echo-reply) exchanges. Prevents comparison of received magic number with internally generated magic number, so that a mismatch does not cause session termination.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set ppp-options ignore-magic-number-mismatch
```

To configure the PPP attributes for statically created si interfaces:

1. Specify the logical inline service interface.

```
[edit interfaces si-slot/pic/port]
user@host# edit unit logical-unit-number
```

2. Configure the interval between PPP keepalive messages for the L2TP tunnel terminating on the LNS.

```
[edit interfaces si-slot/pic/port unit logical-unit-number]
user@host# set keepalives interval seconds
```

3. Configure the number of keepalive packets a destination must fail to receive before the network takes down a link.

```
[edit interfaces si-slot/pic/port unit logical-unit-number]
user@host# set keepalives down-count number
```

NOTE: The `keepalives up-count` option is typically not used for subscriber management.

4. Configure PPP authentication methods that apply to tunneled PPP subscribers at the LNS.

```
[edit interfaces si-slot/pic/port unit logical-unit-number]
user@host# set ppp-options chap
user@host# set ppp-options pap
```

5. Configure the router to prompt the Customer Premises Equipment (CPE) to negotiate both primary and secondary DNS addresses during IPCP negotiation for tunneled PPP subscribers at the LNS.

```
[edit interfaces si-slot/pic/port unit logical-unit-number]
user@host# set ppp-options ipcp-suggest-dns-option
```

BEST PRACTICE: Although all other statements subordinate to `ppp-options`—including those subordinate to `chap` and `pap`—are supported, they are typically not used for subscriber management. We recommend that you leave these other statements at their default values.

NOTE: You can also configure PPP attributes with a user group profile that applies the attributes to all subscribers with that profile on a LAC client. See ["Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile"](#) on page 264 for more information. When you configure the PPP attributes for L2TP LNS subscribers both on the si interface and in user group profiles, the inline service interface configuration takes precedence over the user group profile configuration.

NOTE: When PPP options are configured in both a group profile and a dynamic profile, the dynamic profile configuration takes complete precedence over the group profile when the dynamic profile includes one or more of the PPP options that can be configured in the group profile. Complete precedence means that there is no merging of options between the profiles. The group profile is applied to the subscriber only when the dynamic profile does not include any PPP option available in the group profile.

Applying PPP Attributes to L2TP LNS Subscribers with a User Group Profile

You can configure a user group profile that enables the LNS to apply PPP attributes to the PPP subscribers tunneled from the LAC. The user group profile is associated with clients (LACs) in the L2TP access profile. Consequently all subscribers handled by a given client share the same PPP attributes.

To configure a user group profile:

1. Create the profile.

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

2. Configure the interval between PPP keepalive messages for the L2TP tunnel terminating on the LNS.

```
[edit access group-profile profile-name]
user@host# set ppp keepalive seconds
```

NOTE: Changes to the keepalive interval in a user group profile affect only new L2TP sessions that come up after the change. Existing sessions are not affected.

3. Configure PPP authentication methods that apply to tunneled PPP subscribers at the LNS.

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

4. Specify a set of AAA options that is used for authentication and authorization of tunneled PPP subscribers at the LNS that are logging in by means of the subscriber and AAA contexts that are specified in the AAA options set.

```
[edit access group-profile profile-name]
user@host# set ppp ppp-options aaa-options aaa-options-name
```

The option set is configured with the `aaa-options aaa-options-name` statement at the `[edit access]` hierarchy level.

5. Configure the router to prompt the Customer Premises Equipment (CPE) to negotiate both primary and secondary DNS addresses during IPCP negotiation for tunneled PPP subscribers at the LNS.

```
[edit access group-profile profile-name]
user@host# set ppp ppp-options ipcp-suggest-dns-option
```

6. (Optional) Disable the Packet Forwarding Engine from performing a validation check for PPP magic numbers received from a remote peer in LCP keepalive (Echo-Request/Echo-Reply) exchanges. This prevents PPP from terminating the session when the number does not match the value agreed upon during LCP negotiation. This capability is useful when the remote PPP peers include arbitrary magic numbers in the keepalive packets. Configuring this statement has no effect on LCP magic number negotiation or on the exchange of keepalives when the remote peer magic number is the expected negotiated number.

```
[edit access group-profile profile-name]
user@host# set ppp ppp-options ignore-magic-number-mismatch
```

7. Configure how long the PPP subscriber session can be idle before it is considered to have timed out.

```
[edit access group-profile profile-name]  
user@host# set ppp idle-timeout 200
```

NOTE: You can also configure PPP attributes on a per-interface basis. See ["Applying PPP Attributes to L2TP LNS Subscribers per Inline Service Interface"](#) on page 261 for more information. When you configure the PPP attributes for L2TP LNS subscribers both on the si interface and in user group profiles, the inline service interface configuration takes precedence over the user group profile configuration.

NOTE: When PPP options are configured in both a group profile and a dynamic profile, the dynamic profile configuration takes complete precedence over the group profile when the dynamic profile includes one or more of the PPP options that can be configured in the group profile. Complete precedence means that there is no merging of options between the profiles. The group profile is applied to the subscriber only when the dynamic profile does not include any PPP option available in the group profile.

Configuring an L2TP Access Profile on the LNS

Access profiles define how to validate Layer 2 Tunneling Protocol (L2TP) connections and session requests. Within each L2TP access profile, you configure one or more clients (LACs). The client characteristics are used to authenticate LACs with matching passwords, and to establish attributes of the client tunnel and session. You can configure multiple access profiles and multiple clients within each profile.

To configure an L2TP access profile:

1. Create the access profile.

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

2. Configure characteristics for one or more clients (LACs).

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

NOTE: Except for the special case of the default client, the LAC client name that you configure in the access profile must match the hostname of the LAC. In the case of a Juniper Networks router acting as the LAC, the hostname is configured in the LAC tunnel profile with the `gateway gateway-name` statement at the `[edit access tunnel-profile profile-name tunnel tunnel-id source-gateway]` hierarchy level. Alternatively, the client name can be returned from RADIUS in the attribute, Tunnel-Client-Auth-Id [90].

NOTE: Use `default` as the client name when you want to define a default tunnel client. The default client enables the authentication of multiple LACs with the same secret and L2TP attributes. This behavior is useful when, for example, many new LACs are added to the network, because it enables the LACs to be used without additional LNS profile configuration. Use `default` only on MX Series routers. The equivalent client name on M Series routers is `*`.

3. (Optional) Specify a local access profile that overrides the global access profile and the tunnel group AAA access profile to configure RADIUS server settings for the client.

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp aaa-access-profile
```

4. Configure the LNS to renegotiate the link control protocol (LCP) with the PPP client. tunneled from the client.

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp lcp-renegotiation
```

5. Configure one or more dynamic service profiles to apply services to all subscribers on the LAC. You can optionally pass parameter to the services in the same statement.

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp service-profile profile-name(parameter)&profile-name
```

6. Configure the maximum number of sessions allowed in a tunnel from the client (LAC).

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp maximum-sessions-per-tunnel number
```

7. Configure the LNS to override result codes 4 and 5 with result code 2 in CDN messages it sends to the LAC when the number of L2TP sessions reaches the configured maximum value. Some third-party LACs cannot fail over to another LNS unless the result code has a value of 2.

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp override-result-code session-out-of-resource
```

8. Configure the tunnel password used to authenticate the client (LAC).

```
[edit access profile access-profile-name client client-name]
user@host# set l2tp shared-secret shared-secret
```

9. (Optional) Associate a group profile containing PPP attributes to apply for the PPP sessions being tunneled from this LAC client.

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

NOTE: If `user-group-profile` is modified or deleted, the existing LNS subscribers, which were using this Layer 2 Tunneling Protocol client configuration, go down.

Configuring a AAA Local Access Profile on the LNS

For some LNS tunnels, you might wish to override the access profile configured at the routing instance that hosts the tunnel with a particular RADIUS server configuration. You can configure a local access profile to do so. You can subsequently use the `aaa-access-profile` statement to apply the local access profile to a tunnel group or LAC client.

A local access profile applied to a client overrides a local access profile applied to a tunnel group, which in turn overrides the access profile for the routing instance.

To configure an AAA local access profile:

1. Create the access profile.

```
[edit access]
user@host# edit profile local-aaa-profile-name
```

2. Configure the order of AAA authentication methods.

```
[edit access profile local-aaa-profile-name]
user@host# set authentication-order radius
```

3. Configure the RADIUS server attributes, such as the authentication password.

```
[edit access profile local-aaa-profile-name]
user@host# set radius-server server-address secret password
```

Configuring an Address-Assignment Pool for L2TP LNS with Inline Services

You can configure pools of addresses that can be dynamically assigned to the tunneled PPP subscribers. The pools must be local to the routing instance where the subscriber comes up. The configured pools are supplied in the RADIUS Framed-Pool and Framed-IPv6-Pool attributes. Pools are optional when Framed-IP-Address is sent by RADIUS.

To configure an address-assignment pool, you must specify the name of the pool and configure the addresses for the pool.

You can optionally configure multiple named ranges, or subsets, of addresses within an address-assignment pool. During dynamic address assignment, a client can be assigned an address from a specific named range. To create a named range, you specify a name for the range and define the address range.

NOTE: Be sure to use the address-assignment pools (`address-assignment`) statement rather than the address pools (`address-pool`) statement.

For more information about address assignment pools, see *Address-Assignment Pools Overview* and *Address-Assignment Pool Configuration Overview*.

To configure an IPv4 address-assignment pool for L2TP LNS:

1. Configure the name of the pool and specify the IPv4 family.

```
[edit access]
user@host# edit address-assignment pool pool-name family inet
```

2. Configure the network address and the prefix length of the addresses in the pool.

```
[edit access address-assignment pool pool-name family inet]
user@host# set network ip-prefix</prefix-length>
```

3. Configure the name of the range and the lower and upper boundaries of the addresses in the range.

```
[edit access address-assignment pool pool-name family inet]
user@host# set range range-name low lower-limit high upper-limit
```

For example, to configure an IPv4 address-assignment pool:

```
[edit access]
user@host# edit address-assignment pool lns-v4-pool family inet
[edit access address-assignment pool lns-v4-pool family inet]
user@host# set network 192.168.1.1/16
[edit access address-assignment pool lns-v4-pool family inet]
user@host# set range lns-v4-pool-range low 192.168.1.1 high 192.168.255.255
```

To configure an IPv6 address-assignment pool for L2TP LNS:

1. Configure the name of the pool and specify the IPv6 family.

```
[edit access]
user@host# edit address-assignment pool pool-name family inet6
```

2. Configure the IPv6 network prefix for the address pool. The prefix specification is required when you configure an IPv6 address-assignment pool.

```
[edit access address-assignment pool pool-name family inet6]
user@host# set prefix ipv6-prefix
```

3. Configure the name of the range and define the range. You can define the range based on the lower and upper boundaries of the prefixes in the range, or based on the length of the prefixes in the range.

```
[edit access address-assignment pool pool-name family inet6]
user@host# set range range-name low lower-limit high upper-limit
```

For example, to configure an IPv6 address-assignment pool:

```
[edit access]
user@host# edit address-assignment pool lns-v6-pool family inet6
[edit access address-assignment pool lns-v6-pool family inet6]
user@host# set prefix 2001:DB8::/32
[edit access address-assignment pool lns-v6-pool family inet6]
user@host# set range lns-v6-pool-range low 2001:DB8:1::/48 high 2001:DB8::ffff::/48
```

Configuring the L2TP LNS Peer Interface

The peer interface connects the LNS to the cloud towards the LACs so that IP packets can be exchanged between the tunnel endpoints. MPLS and aggregated Ethernet can also be used to reach the LACs.

NOTE: On MX Series routers, you must configure the peer interface on an MPC.

To configure the LNS peer interface:

1. Specify the interface name.

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

2. Enable VLANs.

```
[edit interfaces interface-name]
user@host# set vlan-tagging
```

- Specify the logical interface, bind a VLAN tag ID to the interface, and configure the address family and the IP address for the logical interface.

```
[edit interfaces interface-name]
user@host# edit unit logical-unit-number
[edit interfaces interface-name unit logical-unit-number]
user@host# set vlan-id number
user@host# set family family address ip-address
```

NOTE: The IPv6 address family is not supported as a tunnel endpoint.

Enabling Inline Service Interfaces

The inline service interface is a virtual physical interface that resides on the Packet Forwarding Engine. This *si* interface, referred to as an *anchor* interface, makes it possible to provide L2TP services without a special services PIC. The inline service interface is supported only by MPCs on MX Series routers. Four inline service interfaces are configurable per MPC-occupied chassis slot.

NOTE: On MX80 and MX104 routers, you can configure only four inline services physical interfaces as anchor interfaces for L2TP LNS sessions: *si-1/0/0*, *si-1/1/0*, *si-1/2/0*, and *si-1/3/0*. You cannot configure *si-0/0/0* for this purpose on MX80 and MX104 routers.

Although the range of bandwidth values is 1 Gbps through 400 Gbps, you cannot configure the bandwidth in absolute numbers such as 12,345,878,000 bps. You must use the options available in the CLI statement:

- 1g
- 10g through 100g in 10 Gbps increments: 10g, 20g, 30g, 40g, 50g, 60g, 70g, 80g, 90g, 100g
- 100g through 400g in 100 Gbps increments: 100g, 200g, 300g, 400g

The maximum bandwidth available varies among MPCs, as shown in [Table 27 on page 273](#). A system log message is generated when you configure a bandwidth higher than is supported on the MPC.

Table 27: Maximum Bandwidth for Inline Services per MPC

MPC	Maximum Supported Bandwidth
MPC2E NG, MPC2E NG Q,	80 Gbps
MPC3E NG, MPC3E NG Q	130 Gbps
100GE and 40GE MPC3 and MICs	40 Gbps
MPC4E	130 Gbps
MPC5E	130 Gbps
MPC6E	130 Gbps
MPC7E	240 Gbps
MPC8E	240 Gbps 400 Gbps in 1.6 Tbps upgraded mode
MPC9E	400 Gbps

To enable inline service interfaces:

1. Access an MPC-occupied slot and the PIC where the interface is to be enabled.

```
[edit chassis]
user@host# edit fpc slot-number pic number
```

2. Enable the interface and optionally specify the amount of bandwidth reserved on each Packet Forwarding Engine for tunnel traffic using inline services. Starting in Junos OS Release 16.2, you are not required to explicitly specify a bandwidth for L2TP LNS tunnel traffic using inline services. When you do not specify a bandwidth, the maximum bandwidth supported on the PIC is automatically

available for the inline services; inline services can use up to this maximum value. In earlier releases, you must specify a bandwidth when you enable inline services with the `inline-services` statement.

```
[edit chassis fpc slot-number pic number]
user@host# set inline-services bandwidth bandwidth-value
```

Configuring an Inline Service Interface for L2TP LNS

The inline service interface is a virtual physical service interface that resides on the Packet Forwarding Engine. This `si` interface, referred to as an *anchor* interface, makes it possible to provide L2TP services without a special services PIC. The inline service interface is supported only by MPCs on MX Series routers. Four inline service interfaces are configurable per MPC-occupied chassis slot.

You can maximize the number of sessions that can be shaped in one service interface by setting the maximum number of hierarchy levels to two. In this case, each LNS session consumes one L3 node in the scheduler hierarchy for shaping.

If you do not specify the number of levels (two is the only option), then the number of LNS sessions that can be shaped on the service interface is limited to the number of L2 nodes, or 4096 sessions. Additional sessions still come up, but they are not shaped.

To configure an inline service interface:

1. Access the service interface.

```
[edit interfaces]
user@host# edit si-slot/pic/port
```

2. (Optional; for per-session shaping only) Enable the inline service interface for hierarchical schedulers and limit the number of scheduler levels to two.

```
[edit interfaces si-slot/pic/port]
user@host# set hierarchical-scheduler maximum-hierarchy-levels 2
```

3. (Optional; for per-session shaping only) Configure services encapsulation for inline service interface.

```
[edit interfaces si-slot/pic/port]
user@host# set encapsulation generic-services
```

4. Configure the IPv4 family on the reserved unit 0 logical interface.

```
[edit interfaces si-slot/pic/port]
user@host# set unit 0 family inet
```

Configuring Options for the LNS Inline Services Logical Interface

You must specify characteristics—*dial-options*—for each of the inline services logical interfaces that you configure for the LNS. LNS on MX Series routers supports only one session per logical interface, so you must configure it as a dedicated interface; the shared option is not supported. (LNS on M Series routers supports dedicated and shared options.) You also configure an identifying name for the logical interface that matches the name you specify in the access profile.

You must specify the *inet* address family for each static logical interface or in the dynamic profile for dynamic LNS interfaces. Although the CLI accepts either *inet* or *inet6* for static logical interfaces, the subscriber cannot log in successfully unless the address family *inet* is configured.

NOTE: For dynamic interface configuration, see "[Configuring a Dynamic Profile for Dynamic LNS Sessions](#)" on page 314.

To configure the static logical interface options:

1. Access the inline services logical interface.

```
[edit]
user@host# edit interfaces si-fpc/pic/port unit logical-unit-number
```

2. Specify an identifier for the logical interface.

```
[edit interfaces si-fpc/pic/port unit logical-unit-number]
user@host# set dial-options l2tp-interface-id name
```

3. Configure the logical interface to be used for only one session at a time.

```
[edit interfaces si-fpc/pic/port unit logical-unit-number]
user@host# set dial-options dedicated
```

4. Configure the address family for each logical interface and enable the local address on the LNS that provides local termination for the L2TP tunnel to be derived from the specified interface name.

```
[edit interfaces si-fpc/pic/port unit logical-unit-number]
user@host# set family inet unnumbered-address lo0.0
```

LNS 1:1 Stateful Redundancy Overview

By default, when an inline service (si) anchor interface goes down—for example, when the card hosting the interface fails or restarts—L2TP subscriber traffic is lost. When the PPP keepalive timer for the tunnel subsequently expires, the control plane goes down and the PPP client is disconnected. Consequently, the client must then reconnect.

You can avoid traffic loss in these circumstances by configuring an aggregated inline service interface (asi) bundle to provide 1:1 stateful redundancy, also called hot standby or active-backup redundancy. The bundle consists of a pair of si physical interfaces, the primary (active) member link and the secondary (standby or backup) member link. These interfaces must be configured on different MPCs; redundancy is not achievable if you configure the primary and secondary interface on the same MPC because both member interfaces go down if the card goes down.

When subscribers log in and 1:1 redundancy is configured, the L2TP session is established over an underlying virtual logical interface (asix.0) over the asi0 physical interface. Individual subscriber logical interfaces are created on the underlying interface in the format, *asiX.logical-unit-number*. The session remains up in the event of a failure or a restart on the MPC hosting the primary member link interface. All the data traffic destined for this L2TP session automatically moves over to the secondary member link interface on the other MPC.

Configuring 1:1 LNS Stateful Redundancy on Aggregated Inline Service Interfaces

You can create an aggregated inline service interface (asi) bundle to provide 1:1 LNS stateful redundancy for inline service (si) anchor interfaces. The bundle pairs two interfaces that reside on different MPCs as primary and secondary links. LNS sessions are subsequently established over a virtual logical interface, *asiX.logical-unit-number*. LNS session failover occurs when either the primary anchor interface goes down or the card is restarted with the `request chassis fpc restart` command. When this happens, the secondary link—on a different MPC—becomes active and all the LNS data traffic destined for the session automatically moves over to the secondary interface. The subscriber session remains up on the

`asiX.logical-unit-number` virtual interface. No traffic statistics are lost. When this redundancy is not configured, subscriber traffic is lost, the keepalives expire, and the PPP client is disconnected and must reconnect.

Before you begin, you must do the following:

- Confirm that enhanced subscriber management is enabled.
- Create inline service interfaces on different MPCs to be aggregated in the bundle.

See ["Enabling Inline Service Interfaces" on page 272](#) and ["Configuring an Inline Service Interface for L2TP LNS" on page 274](#).

- If you are using pools of service interfaces, define the service pools.

BEST PRACTICE: Follow these guidelines:

- You must configure unit 0 family inet for each bundle; otherwise, the session fails to come up.
- The primary (active) and secondary (backup) interfaces must be on different MPCs.
- The bandwidth configured at the `[edit chassis fpc slot pic number inline-services bandwidth]` hierarchy level must be the same for both member links.
- An si interface configured as a member of an aggregated inline service interface bundle cannot be configured as a member of another bundle group.
- An si interface configured as a member of an aggregated inline service interface bundle cannot also be used for any function that is not related to aggregated services; for example, it cannot be used for inline IP reassembly.
- When you configure an si interface as a member of an aggregated inline services bundle, you can no longer configure that si interface independently. You can configure only the parent bundle; the bundle's configuration is applied immediately to all member interfaces.

To configure 1:1 LNS stateful redundancy:

1. On one MPC, specify the primary (active) inline services member link in the bundle.

```
[edit interfaces asix aggregated-inline-services-options]
user@host# set primary-interface
```

2. Configure the amount of bandwidth reserved on this MPC for tunnel traffic using the primary inline service interface.

```
[edit chassis fpc slot pic number inline-services]
user@host# set bandwidth (1g | 10g)
```

3. On a different MPC, specify the secondary(backup) inline services member link in the bundle.

```
[edit interfaces asix aggregated-inline-services-options]
user@host# set secondary-interface
```

NOTE: If you configure the active and backup member links on the same MPC, the subsequent commit of the configuration fails.

4. Configure the amount of bandwidth reserved on this MPC for tunnel traffic using the secondary inline service interface.

```
[edit chassis fpc slot pic number inline-services]
user@host# set bandwidth (1g | 10g)
```

5. Assign the aggregated inline service interface bundle to an L2TP tunnel group by either of the following methods:

- Assign a single bundle by specifying the name of the aggregated inline service physical interface.

```
[edit services l2tp tunnel-group name]
user@host# set service-interface interface-name
```

- Assign one or more pools of bundles to the tunnel group.

```
[edit services l2tp tunnel-group name]
user@host# set service-device-pool pool-name
```

NOTE: A pool can be mixed; that is, it can include both aggregated inline service interface bundles and individual inline service interfaces. The individual interfaces must not be members of existing bundles.

The following sample configuration creates bundle asi0 with member links on MPCs in slot 1 and slot 2, then assigns the bundle to provide redundancy for L2TP sessions on tunnel group tg1:

```
[edit interfaces asi0]
user@host# set aggregated-inline-services-options primary-interface si-1/0/0
user@host# set aggregated-inline-services-options secondary-interface si-2/0/0
user@host# set unit 0 family inet

[edit chassis fpc 1 pic 0 inline-services]
user@host# set bandwidth 10g

[edit chassis fpc 2 pic 0 inline-services]
user@host# set bandwidth 10g

[edit services l2tp tunnel-group tg1]
user@host# set service-interface asi0
```

Verifying LNS Aggregated Inline Service Interface 1:1 Redundancy

IN THIS SECTION

- Purpose | 279
- Action | 279

Purpose

View information about aggregated inline service interface bundles, individual member links, and redundancy status.

Action

- To view summary information about an aggregated inline service interface bundle:

```
user@host> show interfaces asi0 terse
Interface          Admin Link Proto  Local          Remote
```

```

asi0          up    up
asi0.0       up    up    inet

```

- To view detailed information about an aggregated inline service interface bundle:

```

user@host> show interfaces asi0 extensive
Physical interface: asi0, Enabled, Physical link is Up
  Interface index: 223, SNMP ifIndex: 734, Generation: 226
  Type: Adaptive-Services, Link-level type: Adaptive-Services, MTU: 9192, Clocking:
Unspecified, Speed: 20000mbps
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps Internal: 0x4000
  Link type      : Full-Duplex
  Link flags     : None
Physical info   : Unspecified
  Hold-times    : Up 0 ms, Down 0 ms
  Current address: Unspecified, Hardware address: Unspecified
  Alternate link address: Unspecified
  Last flapped  : 2014-01-20 23:35:02 PST (00:03:25 ago)
  Statistics last cleared: Never
Traffic statistics:
  Input bytes   :                0
  Output bytes  :                0
  Input packets:                0
  Output packets:               0
IPv6 transit statistics:
  Input bytes   :                0
  Output bytes  :                0
  Input packets:                0
  Output packets:               0
Input errors:
  Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards: 0,
Resource errors: 0
Output errors:
  Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors: 0

Logical interface asi0.0 (Index 356) (SNMP ifIndex 52241) (Generation 165)
  Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Adaptive-Services
Traffic statistics:
  Input bytes   :                0
  Output bytes  :                0
  Input packets:                0

```

```

Output packets:                0
Local statistics:
Input bytes :                   0
Output bytes :                   0
Input packets:                  0
Output packets:                  0
Transit statistics:
Input bytes :                   0
Output bytes :                   0
Input packets:                  0
Output packets:                  0
Protocol inet, MTU: 9192, Generation: 198, Route table: 0
Flags: Sendbcst-pkt-to-re

```

- To view information about an individual member interface in an aggregated inline service interface bundle:

```

user@host> show interfaces si-1/0/0
Physical interface: si-1/0/0, Enabled, Physical link is Up
Interface index: 165, SNMP ifIndex: 630
Type: Adaptive-Services, Link-level type: Adaptive-Services, MTU: 9192, Speed: 10000mbps
Device flags : Present Running
Interface flags: Point-To-Point SNMP-Traps Internal: 0x4000
Link type : Full-Duplex
Link flags : None
Last flapped : Never
Input rate : 0 bps (0 pps)
Output rate : 0 bps (0 pps)

Logical interface si-1/0/0.0 (Index 357) (SNMP ifIndex 52229)
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: Adaptive-Services
Input packets : 0
Output packets: 0
Protocol asi, AS bundle: asi0.0
Flags: Function2

```

- To view redundancy status for aggregated inline service interface bundles:

```

user@host> show interfaces redundancy
Interface State      Last change Primary   Secondary Current status
asi0      On secondary 1d 23:56  si-1/0/0  si-2/0/0  primary down

```

```

asi1      On primary   10:10:27   si-3/0/0  si-4/0/0  secondary down
ae0       On primary   00:00:02   ge-1/0/0  ge-3/0/1  backup down
ae2       On primary   00:00:01   ge-2/0/0  ge-4/0/1  both up

```

That sample output shows that both aggregated Ethernet and aggregated inline service interfaces are configured for redundancy. To display only one of the aggregated inline service interface bundles:

```

user@host> show interfaces redundancy asi0
Interface State          Last change Primary   Secondary Current status
asi0      On secondary  1d 23:56   si-1/0/0  si-2/0/0  primary down

```

- To view detailed information about all configured redundancy interfaces:

```

user@host> show interfaces redundancy detail
Redundancy interfaces detail
Interface      : asi0
State          : On primary
Last change    : 00:00:36
Primary        : si-1/0/0
Secondary      : si-3/0/0
Current status: both up

Interface      : ae0
State          : On primary
Last change    : 00:01:30
Primary        : ge-1/0/0
Secondary      : ge-3/0/1
Current status : backup down

```

L2TP Session Limits and Load Balancing for Service Interfaces

IN THIS SECTION

- [Session Limits on Service Interfaces | 283](#)
- [Session Load Balancing Across Service Interfaces | 284](#)

The LNS load balances subscriber sessions across the available service interfaces in a device pool based on the number of sessions currently active on the interfaces. You can configure a maximum limit per service interface (si) and per aggregated service interface (asi). In the case of asi interfaces, you cannot configure a limit for the individual si member interfaces in the bundle.

Session Limits on Service Interfaces

When an L2TP session request is initiated for a service interface, the LNS checks the number of current active sessions on that interface against the maximum number of sessions allowed for the individual service interface or aggregated service interface. The LNS determines whether the current session count (displayed by the `show services l2tp summary` command) is less than the configured limit. When that is true or when no limit is configured, the check passes and the session can be established. If the current session count is equal to the configured limit, then the LNS rejects the session request. No subsequent requests can be accepted on that interface until the number of active requests drops below the configured maximum. When a session request is rejected for an si or asi interface, the LNS returns a CDN message with the result code set to 2 and the error code set to 4.

For example, suppose a single service interface is configured in the tunnel group. The current L2TP session count is 1500, with a configured limit of 2000 sessions. When a new session is requested, the limit check passes and the session request is accepted.

Interface	Configured Session Limit	Current Session Count	Session Limit Check Result
si-0/0/0	2000	1500	Pass

The limit check continues to pass and session requests are accepted until 500 requests have been accepted, making the current session count 2000, which matches the configured maximum. The session limit check fails for all subsequent requests and all requests are rejected until the current session count on the interface drops below 2000, so that the limit check can pass.

Interface	Configured Session Limit	Current Session Count	Session Limit Check Result
si-0/0/0	2000	2000	Fail

When the session limit is set to zero for an interface, no session requests can be accepted. If that is the only interface in the tunnel group, then all session requests in the group are rejected until the session limit is increased from zero or another service interface is added to the tunnel group.

When a service interface in a service device pool has reached the maximum configured limit or it has a configured limit of zero, the LNS skips that interface when a session request is made and selects another

interface in the pool to check the session limit. This continues until an interface passes and the session is accepted or no other interface remains in the pool to be selected.

Session Load Balancing Across Service Interfaces

The behavior for session load distribution in a service device pool changed in Junos OS Release 16.2. When a service interface has a lower session count than another interface in the pool and both interfaces are below their maximum session limit, subsequent sessions are distributed to the interface with fewer sessions.

In earlier releases, sessions are distributed in a strictly round-robin manner, regardless of session count. The old behavior can result in uneven session distribution when the Packet Forwarding Engine is rebooted or a service interface goes down and comes back up.

For example, consider the following scenario using the old round-robin distribution behavior for a pool with two service interfaces:

1. Two hundred sessions are evenly distributed across the two service interfaces.
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 100 sessions.
2. The si-1/0/0 interface reboots. When it comes back, initially sessions are up only on si-0/0/0.
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 0 sessions.
3. As the sessions formerly on si-1/0/0 reconnect, they are distributed equally across both service interfaces. When all 100 sessions are back up, the distribution is significantly unbalanced.
 - si-0/0/0 has 150 sessions.
 - si-1/0/0 has 50 sessions.
4. After 100 new sessions connect, si-0/0/0 reaches its maximum limit. Subsequent sessions are accepted only on si-1/0/0.
 - si-0/0/0 has 200 sessions.
 - si-1/0/0 has 100 sessions.
5. After 100 more sessions connect, si-1/0/0 reaches its maximum limit. No more sessions can be accepted until the session count drops below 200 for one of the interfaces.
 - si-0/0/0 has 200 sessions.
 - si-1/0/0 has 200 sessions.

Now consider the same scenario using the current load distribution behavior based on the number of attached sessions. The device pool again has two service interfaces each with a configured maximum limit of 200 sessions:

1. Two hundred sessions are evenly distributed across the two service interfaces.
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 100 sessions.
2. The si-1/0/0 interface reboots. When it comes back up, sessions are up initially only on si-0/0/0.
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 0 sessions.
3. As the sessions formerly on si-1/0/0 reconnect, they are distributed according to the session load on each interface. Because both interfaces are below their maximum limit, and si-1/0/0 has fewer sessions than si-0/0/0, sessions are initially distributed only to si-1/0/0.
 - a. After 1 new session:
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 1 session.
 - b. After 10 new sessions:
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 10 sessions.
 - c. After 100 new sessions:
 - si-0/0/0 has 100 sessions.
 - si-1/0/0 has 100 sessions.
4. Because both interfaces now have the same session count, the next session (#101) is distributed randomly between the two interfaces. The next session after that (#102) goes to the interface with the lower session count. That makes the interfaces equal again, so the next session (#103) is randomly distributed. This pattern repeats until the maximum limit of 200 sessions for both interfaces.
 - si-0/0/0 has 200 sessions.
 - si-1/0/0 has 200 sessions.

No more sessions can be accepted on either interface until the number of sessions drops below 200 on one of the interfaces.

The load balancing behavior is the same for aggregated service interfaces. An asi interface is selected from a pool based on the current session count for the asi interface. When that count is less than the maximum, the LNS checks current session count for the active si interface in the asi bundle. When that count is less than the maximum, the session can be established on the asi interface.

In a mixed device pool that has both service interfaces and aggregated service interfaces, sessions are distributed to the interface, either asi or si, that has the lowest session count. When the session count of an interface of either type reaches its limit, it can no longer accept sessions until the count drops below the maximum.

You can use the session limit configuration to achieve a session limit on particular Packet Forwarding Engines. Suppose you want a limit of 100 sessions on a PFE0, which has two service interfaces. You can set the max limit on each interface to 50, or any other combination that adds up to 100 to establish the PFE0 limit.

Example: Configuring an L2TP LNS

IN THIS SECTION

- [Requirements | 286](#)
- [Overview | 287](#)
- [Configuration | 289](#)

This example shows how you can configure an L2TP LNS on an MX Series router to provide tunnel endpoints for an L2TP LAC in your network. This configuration includes a dynamic profile for dual-stack subscribers.

Requirements

This L2TP LNS example requires the following hardware and software:

- MX Series 5G Universal Routing Platform
- One or more MPCs
- Junos OS Release 11.4 or later

No special configuration beyond device initialization is required before you can configure this feature.

You must configure certain standard RADIUS attributes and Juniper Networks VSAs in the attribute return list on the AAA server associated with the LNS for this example to work. Table 2 lists the attributes with their required order setting and values. We recommend that you use the most current Juniper Networks RADIUS dictionary, available in the *Downloads* box on the *Junos OS Subscriber Management* page at https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/subscriber-access/index.html.

Table 28: VSA and Standard RADIUS Attribute Names, Order, and Values Required for Example

VSA Name [Number]	Order	Value
CoS-Parameter-Type [26-108]	1	T01 Multiplay
CoS-Parameter-Type [26-108]	2	T02 10m
CoS-Parameter-Type [26-108]	3	T08 -36
CoS-Parameter-Type [26-108]	4	T07 cell-mode
Framed-IPv6-Pool [100]	0	jnpr_ipv6_pool
Framed-Pool [88]	0	jnpr_pool
Egress-Policy-Name [26-11]	0	classify
Ingress-Policy-Name [26-10]	0	classify
Virtual-Router [26-1]	0	default

Overview

The LNS employs user group profiles to apply PPP attributes to the PPP subscribers that are tunneled from the LAC. LACs in the network are clients of the LNS. The clients are associated with user group profiles in the L2TP access profile configured on the LNS. In this example, the user group profile `ce-l2tp-group-profile` specifies the following PPP attributes:

- A 30-second interval between PPP keepalive messages for L2TP tunnels from the client LAC terminating on the LNS.

- A 200-second interval that defines how long the PPP subscriber session can be idle before it is considered to have timed out.
- Both PAP and CHAP as the PPP authentication methods that apply to tunneled PPP subscribers at the LNS.

The L2TP access profile `ce-l2tp-profile` defines a set of L2TP parameters for each client LAC. In this example, the user group profile `ce-l2tp-group-profile` is associated with both clients, `lac1` and `lac2`. Both clients are configured to have the LNS renegotiate the link control protocol (LCP) with the PPP client rather than accepting the pre-negotiated LCP parameters that the LACs pass to the LNS. LCP renegotiation also causes authentication to be renegotiated by the LNS; the authentication method is specified in the user group profile. The maximum number of sessions allowed per tunnel is set to 1000 for `lac1` and to 4000 for `lac2`. A different password is configured for each LAC.

A local AAA access profile, `aaa-profile`, enables you to override the global AAA access profile, so that you can specify an authentication order, a RADIUS server that you want to use for L2TP, and a password for the server.

In this example, an address pool defines a range of IP addresses that the LNS allocates to the tunneled PPP sessions. This example defines ranges of IPv4 and IPv6 addresses.

Two inline service interfaces are enabled on the MPC located in slot 5 of the router. For each interface, 10 Gbps of bandwidth is reserved for tunnel traffic on the interface's associated PFE. These *anchor* interfaces serve as the underlying physical interface. To enable CoS queue support on the individual logical inline service interfaces, you must configure both services encapsulation (`generic-services`) and hierarchical scheduling support on the anchors. The IPv4 address family is configured for both anchor interfaces. Both anchor interfaces are specified in the `lns_p1` service device pool. The LNS can balance traffic loads across the two anchor interfaces when the tunnel group includes the pool.

This example uses the dynamic profile `dyn-lns-profile2` to specify characteristics of the L2TP sessions that are created or assigned dynamically when a subscriber is tunneled to the LNS. For many of the characteristics, a predefined variable is set; the variables are dynamically replaced with the appropriate values when a subscriber is tunneled to the LNS.

The interface to which the tunneled PPP client connects (`$junos-interface-name`) is dynamically created in the routing instance (`$junos-routing-instance`) assigned to the subscriber. Routing options for access routes include the route's next hop address (`$junos-framed-route-nexthop`), metric (`$junos-framed-route-cost`), and preference (`$junos-framed-route-distance`). For access-internal routes, a dynamic IP address variable (`$junos-subscriber-ip-address`) is set.

The logical inline service interfaces are defined by the name of a configured anchor interface (`$junos-interface-ifd-name`) and a logical unit number (`$junos-interface-unit`). The profile assigns `l2tp-encapsulation` as the identifier for the logical interface and specifies that each interface can be used for only a single session at a time.

The IPv4 address is set to a value returned from the AAA server. For IPv4 traffic an input firewall filter `$junos-input-filter` and an output firewall filter `$junos-output-filter` are attached to the interface. The loopback variable (`$junos-loopback-interface`) derives an IP address from a loopback interface (`lo`) configured in the routing instance and uses it in IPCP negotiation as the PPP server address. Because this is a dual-stack configuration, the IPv6 address family is also set, with the addresses provided by the `$junos-ipv6-address` variable.

The `$junos-ipv6-address` variable is used because Router Advertisement Protocol is also configured. This variable enables AAA to allocate the first address in the prefix to be reserved as the local address for the interface. The minimal configuration for the Router Advertisement Protocol in the dynamic profile specifies the `$junos-interface-name` and `$junos-ipv6-ndra-prefix` variables to dynamically assign a prefix value in IPv6 neighbor discovery router advertisements.

The dynamic profile also includes the class of service configuration that is applied to the tunnel traffic. The traffic-control profile (`tc-profile`) includes variables for the scheduler map (`$junos-cos-scheduler-map`), shaping rate (`$junos-cos-shaping-rate`), overhead accounting (`$junos-cos-shaping-mode`), and byte adjustment (`$junos-cos-byte-adjust`). The dynamic profile applies the CoS configuration—including the forwarding class, the output traffic-control profile, and the rewrite rules—to the dynamic service interfaces.

The `tg-dynamic` tunnel group configuration specifies the access profile `ce-l2tp-profile`, the local AAA profile `aaa-profile`, and the dynamic profile `dyn-lns-profile2` that are used to dynamically create LNS sessions and define the characteristics of the sessions. The `lns_p1` service device pool associates a pool of service interfaces with the group to enable LNS to balance traffic across the interfaces. The local gateway address `203.0.113.2` corresponds to the remote gateway address that is configured on the LAC. The local gateway name `ce-lns` corresponds to the remote gateway name that is configured on the LAC.

NOTE: This example does not show all possible configuration choices.

Configuration

IN THIS SECTION

- [Procedure | 290](#)

Procedure

CLI Quick Configuration

To quickly configure an L2TP LNS, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
[edit]
edit access group-profile ce-l2tp-group-profile
set ppp idle-timeout 200
set ppp ppp-options pap
set ppp ppp-options chap
set ppp keepalive 30
top
edit access profile ce-l2tp-profile
set client lac1 l2tp maximum-sessions-per-tunnel 1000
set client lac1 l2tp interface-id l2tp-encapsulation-1
set client lac1 l2tp lcp-renegotiation
set client lac1 l2tp shared-secret "lac1-$ABC123"
set client lac1 user-group-profile ce-l2tp-group-profile
set client lac2 l2tp maximum-sessions-per-tunnel 4000
set client lac2 l2tp interface-id l2tp-encap-2
set client lac2 l2tp lcp-renegotiation
set client lac2 l2tp shared-secret "lac2-$ABC123"
set client lac2 user-group-profile ce-l2tp-group-profile
top
edit access profile aaa-profile
set authentication-order radius
set radius authentication-server 198.51.100.193
set radius-server 198.51.100.193 secret "$ABC123"
top
edit access address-assignment pool client-pool1 family inet
set network 192.168.1.1/16
set range lns-v4-pool-range low 192.168.1.1
set range lns-v4-pool-range high 192.168.255.255
top
edit access address-assignment pool client-ipv6-pool2 family inet6
set prefix 2001:DB8::/32
set range lns-v6-pool-range low 2001:DB8:1::/48
set range lns-v6-pool-range high 2001:DB8:ffff::/48
top
set interfaces ge-5/0/1 unit 11 vlan-id 11
```

```
set interfaces ge-5/0/1 unit 11 family inet address 203.0.113.2/24
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
top
set chassis fpc 5 pic 0 inline-services bandwidth 10g
set chassis fpc 5 pic 2 inline-services bandwidth 10g
top
edit interfaces si-5/0/0
set hierarchical-scheduler maximum-hierarchy-levels 2
set encapsulation generic-services
set unit 0 family inet
top
edit interfaces si-5/2/0
set hierarchical-scheduler maximum-hierarchy-levels 2
set encapsulation generic-services
set unit 0 family inet
top
set services service-device-pools pool lns_p1 interface si-5/0/0
set services service-device-pools pool lns_p1 interface si-5/2/0
top
edit dynamic-profiles dyn-lns-profile2 routing-instances $junos-routing-instance
set interface $junos-interface-name
edit routing-options access route $junos-framed-route-ip-address-prefix
set next-hop $junos-framed-route-nexthop
set metric $junos-framed-route-cost
set preference $junos-framed-route-distance
up 2
edit access-internal route $junos-subscriber-ip-address
set qualified-next-hop $junos-interface-name
up 5
edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
set dial-options l2tp-interface-id l2tp-encapsulation
set dial-options dedicated
set family inet filter input $junos-input-filter
set family inet filter output $junos-output-filter
set family inet unnumbered-address $junos-loopback-interface
set family inet6 address $junos-ipv6-address
set family inet6 filter input $junos-input-ipv6-filter
set family inet6 filter output $junos-output-ipv6-filter
up 3
edit protocols router-advertisement
set interface $junos-interface-name prefix $junos-ipv6-ndra-prefix
top
[edit class-of-service]
```

```

edit rewrite-rules dscp rewriteDSCP forwarding-class expedited-forwarding
set loss-priority high code-point af11
set loss-priority high code-point af12
top
edit dynamic-profiles dyn-lns-profile2 class-of-service traffic-control-profiles tc-profile
set scheduler-map $junos-cos-scheduler-map
set shaping-rate $junos-cos-shaping-rate
set overhead-accounting $junos-cos-shaping-mode
set overhead-accounting bytes $junos-cos-byte-adjust
up
edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
set forwarding-class expedited-forwarding
set output-traffic-control-profile tc-profile
set rewrite-rules dscp rewriteDSCP
edit interfaces si-5/0/0
set output-control-profile-remaining tc-profile
top
set services l2tp tunnel-group tg-dynamic l2tp-access-profile ce-l2tp-profile
set services l2tp tunnel-group tg-dynamic aaa-access-profile aaa-profile
set services l2tp tunnel-group tg-dynamic local-gateway address 203.0.113.2
set services l2tp tunnel-group tg-dynamic local-gateway gateway-name ce-lns
set services l2tp tunnel-group tg-dynamic service-device-pool lns_p1
set services l2tp tunnel-group tg-dynamic dynamic-profile dyn-lns-profile2

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an L2TP LNS with inline service interfaces:

1. Configure a user group profile that defines the PPP configuration for tunnel subscribers.

```

[edit access]
user@host# edit group-profile ce-l2tp-group-profile
[edit access group-profile ce-l2tp-group-profile]
user@host# set ppp keepalive 30
user@host# set ppp idle-timeout 200
user@host# set ppp ppp-options chap
user@host# set ppp ppp-options pap

```


2. Configure an L2TP access profile that defines the L2TP parameters for each client LAC. This includes associating a user group profile with the client and specifying the identifier for the inline services logical interface that represents an L2TP session on the LNS.

```
[edit access profile ce-l2tp-profile client lac1]
user@host# set l2tp interface-id l2tp-encapsulation
user@host# set l2tp maximum-sessions-per-tunnel 1000
user@host# set l2tp shared-secret "lac1-$ABC123"
user@host# set l2tp lcp-renegotiation
user@host# set user-group-profile ce-l2tp-group-profile
[edit access profile ce-l2tp-profile client lac2]
user@host# set l2tp interface-id interface-id
user@host# set l2tp maximum-sessions-per-tunnel 4000
user@host# set l2tp shared-secret "lac2-$ABC123"
user@host# set l2tp lcp-renegotiation
user@host# set user-group-profile ce-l2tp-group-profile
```

NOTE: If `user-group-profile` is modified or deleted, the existing LNS subscribers, which were using this Layer 2 Tunneling Protocol client configuration, go down.

3. Configure a AAA access profile to override the global access profile for the order of AAA authentication methods and server attributes.

```
[edit access profile aaa-profile]
user@host# set authentication-order radius
user@host# set radius authentication-server 198.51.100.193
user@host# set radius-server 198.51.100.193 secret "$ABC123"
```

4. Configure IPv4 and IPv6 address-assignment pools to allocate addresses for the clients (LACs).

```
[edit access address-assignment pool client-pool1 family inet]
user@host# set network 192.168.1.1/16
user@host# set range lns-v4-pool-range low 192.168.1.1 high 192.168.255.255
[edit access address-assignment pool client-ipv6-pool2 family inet6]
user@host# set prefix 2001:DB8::/32
user@host# set range lns-v6-pool-range low 2001:DB8:1::/48
user@host# set range lns-v6-pool-range high 2001:DB8:ffff::/48
```

5. Configure the peer interface to terminate the tunnel and the PPP server-side IPCP address (loopback address).

```
[edit interfaces ge-5/0/1
user@host# set vlan-tagging
user@host# set unit 11
[edit interfaces ge-5/0/1.11
user@host# set vlan-id 11
user@host# set family inet address 10.1.1.2/24
[edit interfaces lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
```

6. Enable inline service interfaces on an MPC.

```
[edit chassis fpc 5]
user@host# set pic 0 inline-services bandwidth 10g
user@host# set pic 2 inline-services bandwidth 10g
```

7. Configure the anchor service interfaces with services encapsulation, hierarchical scheduling, and the address family.

```
[edit interfaces si-5/0/0]
user@host# set hierarchical-scheduler maximum hierarchy-levels 2
user@host# set encapsulation generic-services
user@host# set unit 0 family inet
[edit interfaces si-5/2/0]
user@host# set hierarchical-scheduler maximum hierarchy-levels 2
user@host# set encapsulation generic-services
user@host# set unit 0 family inet
```

8. Configure a pool of service interfaces for dynamic LNS sessions.

```
[edit services service-device-pools pool lns_p1]
user@host# set interface si-5/0/0
user@host# set interface si-5/2/0
```

- Configure a dynamic profile that dynamically creates L2TP logical interfaces for dual-stack subscribers.

```
[edit dynamic-profiles dyn-lns-profile2]
user@host# edit routing-instances $junos-routing-instance
user@host# set interface $junos-interface-name
[edit dynamic-profiles dyn-lns-profile2 routing-instances "$junos-routing-instance"]
user@host# edit routing-options access route $junos-framed-route-ip-address-prefix
[edit dynamic-profiles dyn-lns-profile2 routing-instances "$junos-routing-instance" routing-
options access route "$junos-framed-route-ip-address-prefix"]
user@host# set next-hop $junos-framed-route-next-hop
user@host# set metric $junos-framed-route-cost
user@host# set preference $junos-framed-route-distance
[edit dynamic-profiles dyn-lns-profile2 routing-instances "$junos-routing-instance" routing-
options access-internal]
user@host# set route $junos-subscriber-ip-address qualified-next-hop $junos-interface-name
[edit dynamic-profiles dyn-lns-profile2 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set dial-options l2tp-interface-id l2tp-encapsulation
user@host# set dial-options dedicated
user@host# set family inet unnumbered-address $junos-loopback-interface
user@host# set family inet filter input $junos-input-filter
user@host# set family inet filter output $junos-output-filter
user@host# set family inet6 address $junos-ipv6-address
set family inet6 filter input $junos-input-ipv6-filter
set family inet6 filter output $junos-output-ipv6-filter
[edit dynamic-profiles dyn-lns-profile2 protocols router-advertisement]
user@host# set interface $junos-interface-name prefix $junos-ipv6-ndra-prefix
```

- Configure shaping, scheduling, and rewrite rules, and apply in the dynamic profile to tunnel traffic.

```
[edit class-of-service]
user@host# edit rewrite-rules dscp rewriteDSCP forwarding-class expedited-forwarding
user@host# set loss-priority high code-point af11
user@host# set loss-priority high code-point af12
[edit dynamic-profiles dyn-lns-profile2 class-of-service traffic-control-profiles tc-
profile]
user@host# set scheduler-map $junos-cos-scheduler-map
user@host# set shaping-rate $junos-cos-shaping-rate
user@host# set overhead-accounting $junos-cos-shaping-mode
user@host# set overhead-accounting bytes $junos-cos-byte-adjust
```

```
[edit dynamic-profiles dyn-lns-profile2 class-of-service interfaces "$junos-interface-ifd-
name" unit "$junos-interface-unit"]
user@host# set forwarding-class expedited-forwarding
user@host# set output-traffic-control-profile tc-profile
user@host# set rewrite-rules dscp rewriteDSCP
[edit class-of-service interfaces si-5/0/0]
user@host# set output-traffic-control-profile-remaining tc-profile
```

11. Configure the L2TP tunnel group to bring up dynamic LNS sessions using the pool of inline service interfaces to enable load-balancing.

```
[edit services l2tp tunnel-group tg-dynamic]
user@host# set l2tp-access-profile ce-l2tp-profile
user@host# set local-gateway address 10.1.1.2
user@host# set local-gateway gateway-name ce-lns
user@host# set aaa-access-profile aaa-profile
user@host# set dynamic-profile dyn-lns-profile2
user@host# set service-device-pool lns_p1
```

Results

From configuration mode, confirm the access profile, group profile, AAA profile, and address-assignment pools configuration by entering the `show access` command. Confirm the inline services configuration by entering the `show chassis` command. Confirm the interface configuration by entering the `show interfaces` command. Confirm the dynamic profile configuration by entering the `show dynamic-profiles` command. Confirm the tunnel group configuration by entering the `show services l2tp` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show access
group-profile ce-l2tp-group-profile {
  ppp {
    idle-timeout 200;
    ppp-options {
      pap;
      chap;
    }
    keepalive 30;
  }
}
```

```
profile ce-l2tp-profile {
  client lac1 {
    l2tp {
      maximum-sessions-per-tunnel 1000;
      interface-id l2tp-encapsulation-1;
      lcp-renegotiation;
      shared-secret "lac1-$ABC123"; ## SECRET-DATA
    }
    user-group-profile ce-l2tp-group-profile;
  }
  client lac2 {
    l2tp {
      maximum-sessions-per-tunnel 4000;
      interface-id l2tp-encap-2;
      lcp-renegotiation;
      shared-secret "lac2-$ABC123"; ## SECRET-DATA
    }
    user-group-profile ce-l2tp-group-profile;
  }
}
profile aaa-profile {
  authentication-order radius;
  radius-server {
    198.51.100.193 secret "$ABC123"; ## SECRET-DATA
  }
}
address-assignment {
  pool client-pool1 {
    family inet {
      network 192.168.1.1/16;
      range lns-v4-pool-range {
        low 192.168.1.1;
        high 192.168.255.255;
      }
    }
  }
  pool client-ipv6-pool2 {
    family inet6 {
      prefix 2001:DB8::/32;
      range lns-v6-pool-range {
        low 2001:DB8:1::/48;
        high 2001:DB8:ffff::/48;
      }
    }
  }
}
```

```
    }
  }
}

[edit]
user@host# show chassis
fpc 5 {
  pic 0 {
    inline-services {
      bandwidth 10g;
    }
  }
  pic 2 {
    inline-services {
      bandwidth 10g;
    }
  }
}

[edit]
user@host# show interfaces
ge-5/0/1 {
  vlan-tagging;;
  unit 11 {
    vlan-id 11;
    family inet {
      address 203.0.113.2/24;
    }
  }
}
si-5/0/0 {
  hierarchical-scheduler maximum-hierarchy-levels 2;
  encapsulation generic-services;
  unit 0 {
    family inet;
  }
}
si-5/2/0 {
  hierarchical-scheduler maximum-hierarchy-levels 2;
  encapsulation generic-services;
  unit 0 {
```

```

        family inet;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
        }
    }
}

[edit]
user@host# show dynamic-profiles
dyn-lns-profile2 {
    routing-instances {
        "$junos-routing-instance" {
            interface "$junos-interface-name";
            routing-options {
                access {
                    route $junos-framed-route-ip-address-prefix {
                        next-hop "$junos-framed-route-nexthop";
                        metric "$junos-framed-route-cost";
                        preference "$junos-framed-route-distance";
                    }
                }
                access-internal {
                    route $junos-subscriber-ip-address {
                        qualified-next-hop "$junos-interface-name";
                    }
                }
            }
        }
    }
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
            dial-options {
                l2tp-interface-id l2tp-encapsulation;
                dedicated;
            }
            family inet {
                filter {

```

```

        input "$junos-input-filter";
        output "$junos-output-filter";
    }
    unnumbered-address "$junos-loopback-interface";
}
family inet6 {
    address $junos-ipv6-address;
    input $junos-input-ipv6-filter;
    output $junos-output-ipv6-filter;
}
}
}
}
protocols {
    router-advertisement {
        interface "$junos-interface-name" {
            prefix $junos-ipv6-ndra-prefix;
        }
    }
}
class-of-service {
    rewrite-rules {
        dscp rewriteDSCP {
            forwarding-class expedited-forwarding {
                loss-priority high code-point af11
                loss-priority high code-point af12
            }
        }
    }
}
traffic-control-profiles {
    tc-profile {
        scheduler-map "$junos-cos-scheduler-map";
        shaping-rate "$junos-cos-shaping-rate";
        overhead-accounting "$junos-cos-shaping-mode" bytes "$junos-cos-byte-adjust";
    }
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
            forwarding-class expedited-forwarding;
            output-traffic-control-profile tc-profile;
            rewrite-rules {
                dscp rewriteDSCP;
            }
        }
    }
}
}

```



```

    }
  }
}

[edit]
user@host# show services l2tp
tunnel-group tg-dynamic {
  l2tp-access-profile ce-l2tp-profile;
  aaa-access-profile aaa-profile;
  local-gateway {
    address 203.0.113.2;
    gateway-name ce-lns;
  }
  service-device-pool lns_p1;
  dynamic-profile dyn-lns-profile2;
}

```

When you are done configuring the device, enter `commit` from configuration mode.

Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces

The L2TP tunnel group specifies attributes that apply to L2TP tunnels and sessions from a group of LAC clients. These attributes include the access profile used to validate L2TP connection requests made to the LNS on the local gateway address, a local access profile that overrides the global access profile, the keepalive timer, and whether the IP ToS value is reflected.

NOTE: If you delete a tunnel group, all L2TP sessions in that tunnel group are terminated. If you change the value of the `local-gateway-address`, `service-device-pool`, or `service-interface` statements, all L2TP sessions using those settings are terminated. If you change or delete other statements at the `[edit services l2tp tunnel-group name]` hierarchy level, new tunnels you establish use the updated values but existing tunnels and sessions are not affected.

To configure the LNS tunnel group:

1. Create the tunnel group.

```
[edit services l2tp]
user@host# edit tunnel-group group-name
```

NOTE: You can create up to 256 tunnel groups.

2. Specify the service anchor interface responsible for L2TP processing on the LNS.

```
[edit services l2tp tunnel-group name]
user@host# set service-interface interface-name
```

This service anchor interface is required for static LNS sessions, and for dynamic LNS sessions that do not balance traffic across a pool of anchor interfaces. The interface is configured at the [edit interfaces] hierarchy level.

3. (Optional; for load-balancing dynamic LNS sessions only) Specify a pool of inline service anchor interfaces to enable load-balancing of L2TP traffic across the interfaces.

```
[edit services l2tp tunnel-group group-name]
user@host# set service-device-pool pool-name
```

The pool is defined at the [edit services service-device-pools] hierarchy level.

4. (For dynamic LNS sessions only) Specify the name of the dynamic profile that defines and instantiates inline service interfaces for L2TP tunnels

```
[edit services l2tp tunnel-group group-name]
user@host# set dynamic-profile profile-name
```

The profile is defined at the [edit dynamic-profiles] hierarchy level.

5. Specify the access profile that validates all L2TP connection requests to the local gateway address.

```
[edit services l2tp tunnel-group group-name]
user@host# set l2tp-access-profile profile-name
```

6. Configure the local gateway address on the LNS; corresponds to the IP address that is used by LACs to identify the LNS.

```
[edit services l2tp tunnel-group group-name]
user@host# set local-gateway address address
```

7. (Optional) Configure the local gateway name on the LNS, returned in the SCCRP message to the LAC. The name must match the remote gateway name configured on the LAC, or the tunnel cannot be created.

```
[edit services l2tp tunnel-group group-name]
user@host# set local-gateway gateway-name gateway-name
```

8. (Optional) Configure the interval at which the LNS sends hello messages if it has received no messages from the LAC.

```
[edit services l2tp tunnel-group group-name]
user@host# set hello-interval seconds
```

9. (Optional) Specify a local access profile that overrides the global access profile to configure RADIUS server settings for the tunnel group.

```
[edit services l2tp tunnel-group group-name]
user@host# set aaa-access-profile profile-name
```

This local profile is configured at the [edit access profile] hierarchy level.

10. (Optional) Configure the LNS to reflect the IP ToS value from the inner IP header to the outer IP header (applies to CoS configurations).

```
[edit services l2tp tunnel-group group-name]
user@host# set tos-reflect
```

11. (Optional) Specify a dynamic service profile to be applied to the L2TP session at login, along with any parameters to pass to the service.

```
[edit services l2tp tunnel-group group-name]
user@host# set service-profile profile-name(parameter)&profile-name
```

Applying Services to an L2TP Session Without Using RADIUS

Services are applied to L2TP sessions for activation or later modified by vendor-specific attributes (VSAs) from the RADIUS server or in RADIUS Change of Authorization (CoA) requests. Starting in Junos OS Release 18.1R1, you can apply services to L2TP sessions by means of dynamic service profiles without involving RADIUS. In multivendor environments, customers might use only standard RADIUS attributes to simplify management by avoiding the use of VSAs from multiple vendors. However, this complicates the application of services to L2TP sessions because VSAs are generally required to apply services. Local dynamic service profile activation enables you to avoid that problem. You can also use local service profile activation to provide default services when RADIUS servers are down.

You can apply services to all subscribers in a tunnel group or to all subscribers using a particular LAC. You can configure a maximum of 12 services per tunnel group or LAC hostname.

After configuring one or more dynamic service profiles that define services, you apply them in the tunnel group or in the access profile configuration for a LAC client by specifying the service profile names. You can list more than one profile to be activated, separated by an ampersand (&). You can also specify parameters to be used by the service profile that might override values configured in the profile itself, such as a downstream shaping rate for a CoS service.

The locally configured list of services (via service profiles) serves as local authorization that is applied by `authd` during client session activation. This list of services is subject to the same validation and processing as services originating from external authority, such as RADIUS. These services are presented during subscriber login.

You can still use RADIUS VSAs or CoA requests in concert with the service profiles. If services are sourced from an external authority as authorization during authentication or during subscriber session provisioning (activation), the services from the external authority take strict priority over those in the local configuration. If a service applied with RADIUS is the same as a service applied with a service profile in the CLI, but with different parameters, the RADIUS service is applied with a new session ID and takes precedence over the earlier service profile.

You can issue commands to deactivate or reactivate any service you have previously activated for a tunnel group or LAC.

Define the dynamic service profiles that you want to later apply to a tunnel group or LAC.

To apply service profiles to all subscribers in a tunnel group:

- Specify one or more service profiles and any parameters to be passed to the services.

```
[edit services l2tp tunnel-group group-name]  
user@host# set service-profile profile-name(parameter)&profile-name
```

To apply service profiles to all subscribers for a particular LAC:

- Specify one or more service profiles and any parameters to be passed to the services.

```
[edit access profile profile-name client client-name l2tp]
user@host# set service-profile profile-name(parameter)&profile-name
```

NOTE: When service profiles are configured for a LAC client and for a tunnel group that uses that client, only the LAC client service profile is applied. It overrides the tunnel group configuration. For example, in the following configuration, the tunnel group, tg-LAC-3, uses the LAC client, LAC-3, so the LAC3 configuration overrides the tunnel group configuration. Consequently only the cos-A3 service is activated for subscribers in the tunnel group, rather than Cos2 and fw1. The shaping rate passed for the service is 24 Mbps.

```
[edit]
user@host# set services l2tp tunnel-group tg-LAC-3 service-profile cos2(31000000)&fw1
user@host# set access profile prof-lac client LAC-3 l2tp service-profile cos-A3(24000000)
```

You can deactivate any service applied to a subscriber session by issuing the following command:

```
user@host> request network-access aaa subscriber delete session-id subscriber-session-id
service-profile profile-name
```

You can reactivate any service applied to a subscriber session by issuing the following command:

```
user@host> request network-access aaa subscriber add session-id subscriber-session-id service-
profile profile-name
```

To display the services sessions for all current subscriber sessions, use the `show subscribers extensive` or `show network-access aaa subscribers session-id id-number detail` command.

To understand how local service application works, the following examples illustrate the various configuration possibilities. First, consider the following dynamic service profile configurations, `cos2` and `fw1`:

```
dynamic-profiles {
  cos2 {
    variables {
```

```

shaping-rate default-value 10m;
shaping-rate-in default-value 10m;
data-in-filter uid;
data-in-policer uid;
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
            family inet;
        }
    }
}
class-of-service {
    traffic-control-profiles {
        TrafficShaper {
            scheduler-map a;
            shaping-rate "$shaping-rate";
        }
    }
    interfaces {
        "$junos-interface-ifd-name" {
            unit "$junos-interface-unit" {
                output-traffic-control-profile TrafficShaper;
            }
        }
    }
}
}
|

```

```

dynamic-profiles {
    fw1 {
        variables {
            v6input default-value v6ingress;
            v6output default-value v6egress;
            input default-value upstrm-filter;
            output default-value dwnstrm-filter;
        }
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-interface-unit" {

```

```

        family inet;
    }
}
}
}
}

```

The following statement applies both services to all subscribers in tunnel group tg1; a parameter value of 31 Mbps is passed to the cos2 service:

```

[edit]
user@host# set services l2tp tunnel-group tg1 service-profile cos2(31000000)&fw1

```

In the cos2 service profile, the shaping rate is provided by a user-defined variable with a default value of 10m, or 1Mbps. After the L2TP session is up, cos2 and fw1 are activated with service session IDs of 34 and 35, respectively.

```

user@host1> show subscribers extensive
...

Service Session ID: 34
  Service Session Name: cos2
  State: Active
  Family: inet
  Service Activation time: 2018-02-15 15:44:16 IST

Service Session ID: 35
  Service Session Name: fw1
  State: Active
  Family: inet
  Service Activation time: 2018-02-15 15:44:16 IST
  Dynamic configuration:
    input: upstrm-filter
    output: dnstrm-filter
    v6input: v6ingress
    v6output: v6egress

```

The parameter passed to cos2 is used as the value for \$shaping-rate; consequently the shaping rate for the service is adjusted from the default value of 10 Mbps to 31 Mbps, as shown in the following command output. Although the output indicates the adjusting application is RADIUS CoA, the

adjustment is a consequence of the parameter passed to the service profile. That operation uses the same internal framework as a CoA and is reported as such.

```
user@host1> show class-of-service interface si-1/0/0.3221225492
  Logical interface: si-1/0/0.3221225492, Index: 3221225492
Object          Name                Type                Index
Traffic-control-profile subscriber-tcp-2      Output              23571
Scheduler-map   a                   Output              4294967354
Classifier       dscp-ipv6-compatibility dscp-ipv6          9
Classifier       ipprec-compatibility ip                  13
```

Adjusting application: RADIUS CoA

```
Adjustment type: absolute
configured-shaping-rate: 31000000
adjustment-value: 31000000
Adjustment overhead-accounting mode: frame mode
Adjustment overhead bytes: 0
Adjustment target: node
Adjustment priority: 1
```

Now the cos2 service is deactivated from the CLI for subscriber session 27.

```
user@host1> request network-access aaa subscriber delete service-profile cos2 session-id 27
Successful completion
```

The following output shows cos2 is gone, leaving only fw1 as an active service.

```
user@host1> show subscribers extensive
Type: L2TP
User Name: user@example.com
IP Address: 192.0.2.103
IP Netmask: 255.255.255.255
Logical System: default
Routing Instance: default
Interface: si-1/0/0.3221225492
Interface type: Dynamic
Underlying Interface: si-1/0/0.3221225492
Dynamic Profile Name: dyn-lns-profile
State: Active
Radius Accounting ID: 27
Session ID: 27
```



```

PFE Flow ID: 42
Login Time: 2017-08-30 07:29:39 IST
Service Sessions: 1
IP Address Pool: ipv4_pool
Accounting interval: 600
Frame/cell mode: Frame
Overhead accounting bytes: -38
Calculated downstream data rate: 1000000 kbps
Adjusted downstream data rate: 1000000 kbps

```

```

Service Session ID: 35
Service Session Name: fw1
State: Active
Family: inet
Service Activation time: 2018-02-15 15:44:16 IST
Dynamic configuration:
  input: upstrm-filter
  output: dwnstrm-filter
  v6input: v6ingress
  v6output: v6egress

```

The following command reactivates cos2 for subscriber session 27.

```

user@host1> request network-access aaa subscriber add service-profile cos2 session-id 27
Successful completion

```

The reactivated cos2 service has a new service session ID of 36.

```

user@host1> show subscribers extensive
...
Service Session ID: 35
Service Session Name: fw1
State: Active
Family: inet
Service Activation time: 2018-02-15 15:44:16 IST
Dynamic configuration:
  input: upstrm-filter
  output: dwnstrm-filter
  v6input: v6ingress
  v6output: v6egress

```

Service Session ID: 36**Service Session Name: cos2**

State: Active

Family: inet

Service Activation time: 2018-02-15 15:58:23 IST

The reactivated cos2 service uses the default shaping rate, 10 Mbps, from the service profile.

```
user@host1> show class-of-service interface si-1/0/0.3221225492
Logical interface: si-1/0/0.3221225492, Index: 3221225492
Object          Name                Type                Index
Traffic-control-profile subscriber-tcp-2      Output              23571
Scheduler-map   a                   Output              4294967354
Classifier       dscp-ipv6-compatibility dscp-ipv6          9
Classifier       ipprec-compatibility ip                  13
```

Adjusting application: RADIUS CoA

Adjustment type: absolute

configured-shaping-rate: 10000000

adjustment-value: 10000000

Adjustment overhead-accounting mode: frame mode

Adjustment overhead bytes: 0

Adjustment target: node

Adjustment priority: 1

Next, a RADIUS CoA request is received, which includes the Activate-Service VSA (26-65). The VSA specifies and activates the service and specifies a change in the shaping rate of cos2 from the default 10 Mbps to 12 Mbps. The cos2 service session 36 still appears in the output, but is superseded by the new service session initiated by the CoA, 49.

```
user@host1> show subscribers extensive
...
Service Session ID: 35
Service Session Name: fw1
State: Active
Family: inet
Service Activation time: 2018-02-15 15:44:16 IST
Dynamic configuration:
  input: upstrm-filter
  output: dwnstrm-filter
  v6input: v6ingress
```

```
v6output: v6egress
```

Service Session ID: 36

Service Session Name: cos2

State: Active

Family: inet

Service Activation time: 2018-02-15 15:58:23 IST

Service Session ID: 49

Service Session Name: cos2

State: Active

Family: inet

Service Activation time: 2018-02-15 16:25:04 IST

Dynamic configuration:

shaping-rate: 12000000

shaping-rate-in: 10m

```
user@host1> show class-of-service interface si-1/0/0.3221225492
```

```
Logical interface: si-1/0/0.3221225492, Index: 3221225492
```

Object	Name	Type	Index
Traffic-control-profile	subscriber-tcp-2	Output	23571
Scheduler-map	a	Output	4294967354
Classifier	dscp-ipv6-compatibility	dscp-ipv6	9
Classifier	ipprec-compatibility	ip	13

Adjusting application: RADIUS CoA

Adjustment type: absolute

configured-shaping-rate: 12000000

adjustment-value: 12000000

Adjustment overhead-accounting mode: frame mode

Adjustment overhead bytes: 0

Adjustment target: node

Adjustment priority: 1

When a service is applied by both the CLI configuration and a RADIUS VSA (26-65), but with different parameters, the RADIUS configuration overrides the CLI configuration. In the following example, the CLI configuration applies the cos2 service profile with a value of 31 Mbps for the shaping rate.

```
[edit]
```

```
user@host# set services l2tp tunnel-group tg1 service-profile cos2(31000000)
```

The RADIUS Access-Accept message service activation VSA (26-65) applies cos2 with a value of 21 Mbps for the shaping rate.

```
l2tp@l2tp.com  User-Password := "bras"
  Auth-Type = Local,
  Service-Type = Framed-User,
  Framed-Protocol = PPP,
  ERX-Service-Activate:1 += 'cos2(21000000)',
```

The CLI configuration activates service session 22 with a shaping rate of 31 Mbps. The RADIUS VSA activates service session 23 with a shaping rate of 21 Mbps.

```
user@host1> show subscribers extensive
...
Service Session ID: 22
Service Session Name: cos2
State: Active
Family: inet
Service Activation time: 2018-02-16 08:22:03 IST
Dynamic configuration:
  shaping-rate: 31000000
  shaping-rate-in: 10m

Service Session ID: 23
Service Session Name: cos2
State: Active
Family: inet
Service Activation time: 2018-02-16 08:22:03 IST
Dynamic configuration:
  shaping-rate: 21000000
  shaping-rate-in: 10m
```

```
user@host1> show class-of-service interface si-1/0/0.3221225492
Logical interface: si-1/0/0.3221225492, Index: 3221225492
Object          Name                Type                Index
Traffic-control-profile subscriber-tcp-2      Output              23571
Scheduler-map   a                   Output              4294967354
Classifier      dscp-ipv6-compatibility dscp-ipv6          9
Classifier      ipprec-compatibility ip                   13
```

Adjusting application: RADIUS CoA

```

Adjustment type: absolute
configured-shaping-rate: 21000000
adjustment-value: 21000000
Adjustment overhead-accounting mode: frame mode
Adjustment overhead bytes: 0
Adjustment target: node
Adjustment priority: 1

```

Configuring a Pool of Inline Services Interfaces for Dynamic LNS Sessions

You can create a pool of inline service interfaces, also known as a *service device pool*, to enable load-balancing of L2TP traffic across the interfaces. The pool is supported for dynamic LNS configurations, where it provides a set of logical interfaces that can be dynamically created and allocated to L2TP sessions on the LNS. The pool is assigned to an LNS tunnel group. L2TP maintains the state of each inline service interface and uses a round-robin method to evenly distribute the load among available interfaces when new session requests are accepted.

NOTE: Load balancing is available only for dynamically created subscriber interfaces.

LNS sessions anchored on an MPC are not affected by a MIC failure as long as some other path to the peer LACs exists. If the MPC hosting the peer interface fails and there is no path to peer LACs, the failure initiates termination and clean-up of all the sessions on the MPC.

If the MPC anchoring the LNS sessions itself fails, the Routing Engine does not relocate sessions to another slot and all sessions are terminated immediately. New sessions can come up on another available interface when the client retries.

To configure the service device pool:

1. Create the pool.

```

[edit services service-device-pools]
user@host# edit pool pool-name

```

2. Specify the inline service interfaces that make up the pool.

```
[edit services service-device-pools pool pool-name]
user@host# set interface service-interface-name
user@host# set interface service-interface-name
```

Configuring a Dynamic Profile for Dynamic LNS Sessions

You can configure L2TP to dynamically assign inline service interfaces for L2TP tunnels. You must define one or more dynamic profiles and assign a profile to each tunnel group. The LNS supports IPv4-only, IPv6-only, and dual-stack IPv4/IPv6 sessions.

To configure the L2TP dynamic profile:

1. Create the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Configure the interface to be dynamically assigned to the routing instance used by the tunneled PPP clients.

```
[edit dynamic-profiles profile-name routing-instances "$junos-routing-instance"]
user@host# set interface $junos-interface-name
```

3. Configure the routing options for access routes in the routing instance.

```
[edit dynamic-profiles profile-name routing-instances "$junos-routing-instance" routing-
options access]
user@host# set route next-hop $junos-framed-route-nexthop
user@host# set route metric $junos-framed-route-cost
user@host# set route preference $junos-framed-route-distance
```

4. Configure the routing options for access-internal routes in the routing instance.

```
[edit dynamic-profiles profile-name routing-instances "$junos-routing-instance" routing-
options access-internal]
user@host# set route $junos-subscriber-ip-address
```

5. Define the interfaces used by the dynamic profile. The variable is dynamically replaced by one of the configured inline service interfaces.

```
[edit dynamic-profiles profile-name]
user@host# set interfaces $junos-interface-ifd-name
```

6. Configure the inline services logical interfaces to be dynamically instantiated.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name"]
user@host# set unit $junos-interface-unit
```

7. Specify an identifier for the logical interfaces.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit]
user@host# set dial-options l2tp-interface-id name
```

8. Configure each logical interface to be used for only one session at a time.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit]
user@host# set dial-options dedicated
```

9. Configure the address family for the logical interfaces and enable the local address on the LNS that provides local termination for the L2TP tunnel to be derived from the specified interface name.

NOTE: Dynamic LNS sessions require you to include the `dial-options` statement in the dynamic profile, which in turn requires you to include the `family inet` statement. This has the following consequences:

- You must always configure `family inet` regardless of whether you configure IPv4-only, IPv6-only, or dual-stack interfaces in the profile.
- When you configure IPv4-only interfaces, you configure only `family inet` and you must configure the interface address under `family inet`.
- When you configure IPv6-only interfaces, you must also configure `family inet6` and you must configure the interface address under `family inet6`. You do not configure the address under `family inet`.
- When you configure dual-stack, IPv4/IPv6 interfaces, you configure both `family inet` and `family inet6` and an interface address under each family.

For IPv4-only interfaces:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit]
user@host# set family inet unnumbered-address $junos-loopback-interface
```

For IPv6-only interfaces:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit]
user@host# set family inet
user@host# set family inet6 unnumbered-address $junos-loopback-interface
```

For dual-stack IPv4/IPv6 interfaces:

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit]
user@host# set family inet unnumbered-address $junos-loopback-interface
user@host# set family inet6 unnumbered-address $junos-loopback-interface
```

NOTE: If Router Advertisement Protocol is configured, then you configure a numbered address rather than an unnumbered address for the IPv6 local address:

```
user@host# set family inet6 address $junos-ipv6-address
```


See [Broadband Subscriber Sessions User Guide](#) for information about using variables for IPv6-only and dual-stack addressing in dynamic profiles.

Release History Table

Release	Description
18.1R1	Starting in Junos OS Release 18.1R1, you can apply services to L2TP sessions by means of dynamic service profiles without involving RADIUS.
16.2R1	Starting in Junos OS Release 16.2, you are not required to explicitly specify a bandwidth for L2TP LNS tunnel traffic using inline services.

RELATED DOCUMENTATION

[L2TP for Subscriber Access Overview | 144](#)

[Ensuring IPCP Negotiation for Primary and Secondary DNS Addresses | 134](#)

[L2TP Session Limits Overview | 200](#)

Local and Remote Service Activation and Deactivation Using the CLI

[Junos OS Feature Licenses](#)

Session Options for Subscriber Access

IP Packet Reassembly on Inline Service Interfaces

IN THIS SECTION

- [IP Packet Fragment Reassembly for L2TP Overview | 318](#)
- [Configuring IP Inline Reassembly for L2TP | 320](#)

IP Packet Fragment Reassembly for L2TP Overview

You can configure inline service interfaces on MX Series routers with MPCs to support reassembly of fragmented IP packets for an L2TP connection. When packets are transmitted over an L2TP connection, the packets may be fragmented during transmission and need to be reassembled before they are processed further. Efficient reassembly is important for network throughput, scalability, and graceful response to congestion.

Fragmentation of IP packets for transmission and the need to reassemble the IP packets at a destination is a feature of how Layer 2 (the frame layer) and Layer 3 (the packet layer) operate. The maximum size of a frame, set by the Maximum Transmission Unit (MTU) value, and the maximum size of a packet are determined independently. Typically the packet size can far exceed the MTU size defined for the outgoing connection. If the packet size (data plus IP and other headers) exceeds the configured frame size (usually set by the transport medium limits), the packet must be fragmented and split across multiple frames for transmission.

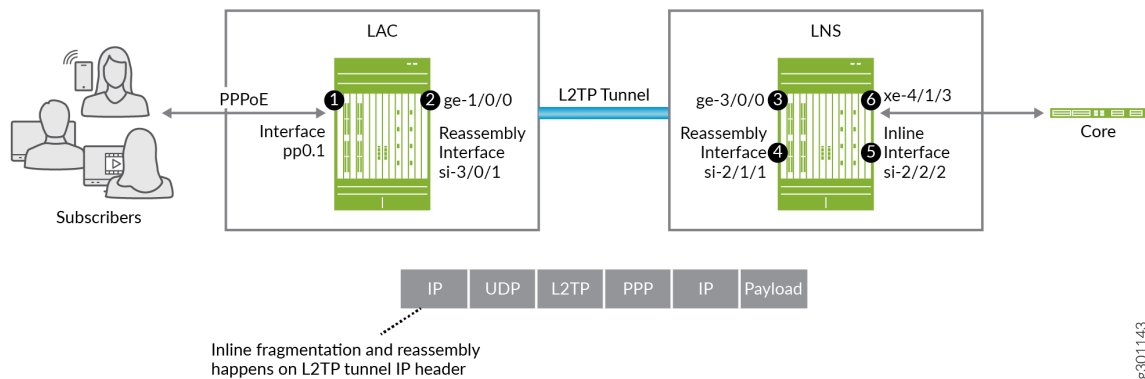
Frames are always processed immediately, when they arrive (if error-free), but packet fragments cannot be processed until the whole packet has been reassembled. Each packet fragment inside a frame series, except the last packet fragment, has the more fragments (MF) IP header bit set, indicating that this packet is part of a whole. The last packet fragment inside a frame does not have this MF bit set and therefore ends the fragment sequence. After all of the fragments of a packet have arrived, the entire packet can be reassembled.

In an L2TP connection, packets are transmitted between the L2TP Access Concentrator (LAC) and the L2TP Network Server (LNS). For an IP packet being transmitted over an L2TP connection, the packet is fragmented at one of the following locations:

- At the LAC for traffic destined for the LNS
- At the LNS for traffic destined for the LAC
- At an intermediate router when the LAC and LNS are not directly connected and the MTU size on the router is less than that on the LAC or LNS.

IP reassembly parameters configured on inline service interfaces of the LAC and the LNS determine how the fragments are reassembled on these interfaces to ensure efficient reassembly over an L2TP connection. [Figure 17 on page 319](#) shows IP fragmentation and reassembly for inbound subscriber traffic in a simplified L2TP network.

Figure 17: L2TP Reassembly for Inbound Traffic

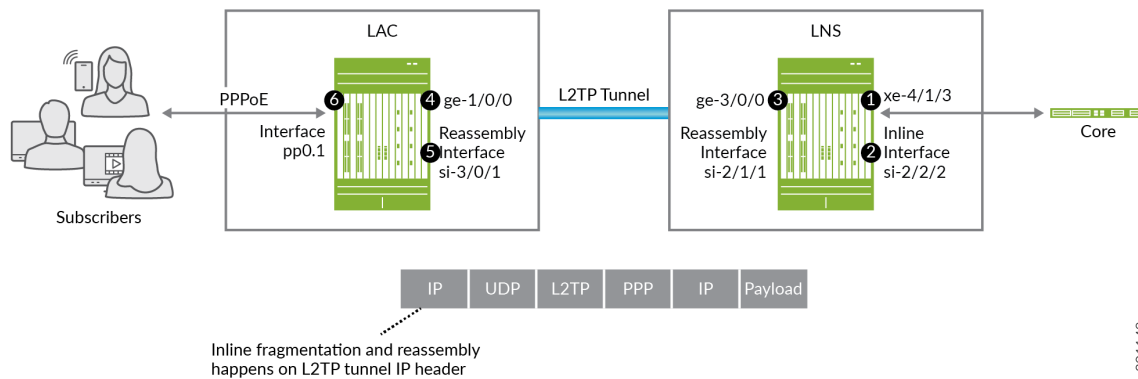


Traffic inbound to the core:

1. Subscriber traffic arrives on the LAC subscriber-facing interface, pp0.1, in the packet format [MAC] [PPPoE] [PPP] [IP] [Payload]. The PPPoE header is stripped off and the L2TP tunnel header is added, creating the tunnel packet, [IP] [UDP] [L2TP] [PPP] [IP] [Payload].
2. The packet is sent out the LAC's peer WAN interface, ge-1/0/0, on the L2TP tunnel. If the packet size is larger than the MTU of the WAN interface, the packet is fragmented on the L2TP tunnel header. The LAC then sends the fragments to the LNS.
3. When the fragments arrive on the LNS peer WAN interface, ge-3/0/0, a route lookup steers the fragments to the LNS reassembly inline interface, si-2/1/1.
4. The fragments are reassembled on interface si-2/1/1 and the packet is sent to the LNS inline interface, si-2/2/2.
5. L2TP decapsulates the L2TP tunnel header and the PPP header on the si-2/2/2 inline interface, leaving the IP header and the payload. A route lookup on the IP header sends the packet to the LNS's core-facing interface, xe-4/1/3.
6. The packet is sent out the core-facing interface, xe-4/1/3.

Figure 18 on page 320 shows IP fragmentation and reassembly for outbound subscriber traffic in a simplified L2TP network.

Figure 18: L2TP Reassembly for Outbound Traffic



Traffic outbound to subscribers:

1. Subscriber traffic arrives on the LNS core-facing interface, xe-4/1/3. A route lookup steers the packet to the LNS inline interface, si-2/2/2.
2. On interface si-2/2/2, L2TP encapsulates the packet with the L2TP header and PPP header, and then creates the L2TP tunnel packet, [IP] [UDP] [L2TP] [PPP] [IP] [Payload].
3. Route lookup on the L2TP tunnel IP header sends the packet to the LNS's peer WAN interface, ge-3/0/0. If the packet size is larger than the MTU of the WAN interface, the packet is fragmented on the L2TP tunnel header. The LNS then sends the fragments to the LAC.
4. When the fragments arrive on the LAC peer WAN interface, ge-1/0/0, a route lookup steers the fragments to the LAC reassembly inline interface, si-3/0/1.
5. The fragments are reassembled on this interface and the packet is sent to the subscriber-facing interface, pp0.1.
6. L2TP decapsulates the L2TP tunnel header on the pp0.1 inline interface, leaving [PPP] [IP] [Payload]. Then PPPoE and MAC encapsulation takes place on the packet. The packet, now consisting of [MAC] [PPPoE] [PPP] [IP] [Payload] is sent out the access interface to the subscriber.

Configuring IP Inline Reassembly for L2TP

This procedure shows how to configure a service interface on a LAC or LNS to reassemble fragmented IP packets. This example creates a service set that configures the IP reassembly parameters for L2TP fragments. The service set is then associated with the L2TP service.

Before you configure inline IP reassembly, be sure you have:

- Configured L2TP.
- Configured a valid service interface on the LAC or LNS.

To configure inline IP reassembly:

1. Configure the chassis-level bandwidth used by the inline services interface on the FPC and PIC slot for inline IP fragment reassembly.

```
[edit chassis]
user@host# set fpc 2 pic 1 inline-services bandwidth 10g
```

2. Configure the interface-level logical unit used by the inline services (si-) interface on the FPC and PIC slot for inline IP fragment reassembly.

```
[edit interfaces]
user@host# set si-2/1/0 unit 0 family inet
user@host# set si-2/1/0 unit 0 service-domain inside
```

NOTE: This configuration is not unique to L2TP. However, you must configure the family (inet) and service domain (inside) as shown.

3. Configure the service set (set1) for IP reassembly in the input match direction. (The local option loops the reassembled packets back to the local interface.)

```
[edit services]
user@host# set service-set set1
[edit services service-set set1]
user@host# set ip-reassembly-rules ipr_rule1
user@host# set next-hop-service inside-service-interface si-2/1/0.0
user@host# set next-hop-service outside-service-interface-type local
```

NOTE:

- You must configure both inside (si- interface) and outside type (local) service interfaces statements. The reassembly rule is not formulated outside of the service set; this statement simply initiates the reassembly process.

- You can configure only one service interface for each service-set.

4. Configure the IP reassembly rule parameter.

```
[edit services ip-reassembly]
user@host# set rule ipr_rule1 match-direction input;
```

5. Configure the service set (set1) for IP reassembly to bind to the L2TP service.

NOTE:

- The service set must be defined at the [edit services] hierarchy level.
- You cannot delete a service set instance if it is associated with an L2TP service.

```
[edit services l2tp]
user@host# set ip-reassembly service-set set1
```

RELATED DOCUMENTATION

[Configuring an L2TP LNS with Inline Service Interfaces | 259](#)

[Protocols and Applications Supported on the MPC1E for MX Series Routers](#)

Peer Resynchronization After an L2TP Failover

IN THIS SECTION

- [L2TP Failover and Peer Resynchronization | 323](#)
- [Configuring the L2TP Peer Resynchronization Method | 324](#)

L2TP Failover and Peer Resynchronization

L2TP failover enables a failed L2TP endpoint to resynchronize with its nonfailed peer during recovery and restart of the L2TP protocol on the failed endpoint. L2TP failover is enabled by default.

The failover and L2TP peer resynchronization process does all of the following:

- Prevents the nonfailed endpoint from prematurely terminating a tunnel while the failed endpoint is recovering.
- Reestablishes the sequence numbers required for the operation of the L2TP control protocol.
- Resolves inconsistencies in the tunnel and session databases of the failed endpoint and the nonfailed endpoint.

The router supports both the L2TP failover protocol method (described in *RFC 4951, Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"*) and the L2TP silent failover method. The differences between these two methods are as follows:

- The L2TP failover protocol method requires a nonfailed endpoint to wait an additional recovery time period while the failed endpoint is recovering to prevent the nonfailed endpoint from prematurely disconnecting the tunnel. The additional recovery period delays the detection of tunnel keepalive failures.

If a peer on an MX series router negotiates failover protocol with an MX Series peer that is not configured for failover protocol, both use the silent failover method. If the negotiation is with a third-party device that does not support failover protocol, the MX Series peer falls back to silent failover; whether the third-party peer recovers in this case depends on how resynchronization is implemented on that device.

- Silent failover operates entirely within the failed endpoint and does not require nonfailed endpoint support—this improves interoperability between peers. Silent failover does not require additional recovery time by the nonfailed endpoint, which also eliminates the potential for degraded responsiveness to the loss of tunnel connectivity. Starting in Junos OS Release 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, silent failover is the default resynchronization method in Junos OS.

In lower-numbered releases, the default resynchronization method is *failover-protocol-fall-back-to-silent-failover*. The recovery method used depends on the results of the failover capability negotiation that takes place between L2TP peers when they establish a tunnel, which works as follows:

- L2TP on the LAC by default attempts to negotiate the L2TP failover protocol first. When L2TP determines that the remote peer supports the L2TP failover protocol, then the L2TP failover protocol method is used.

- When L2TP determines that the remote peer does not support the L2TP failover protocol, then the L2TP silent failover method is used. Falling back on this secondary method prevents the failover from forcing a disconnection of the tunnel to the peer and all its sessions.

In Junos OS releases where `failover-protocol-fall-back-to-silent-failover` is the default method, you can change the default behavior by including the `disable-failover-protocol` statement at the `[edit services l2tp]` hierarchy level. This statement forces the configured LAC or LNS endpoint to operate only in silent failover mode. This configuration can be used to prevent the device from negotiating failover protocol with the peer even if the peer tries to negotiate it. When you issue this statement and the peer supports only failover protocol, the nonfailed endpoint (LAC or LNS) uses silent failover for recovery. Starting in Junos OS Release 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, the `disable-failover-protocol` statement is deprecated, because the change in default resynchronization method makes it unnecessary.

Configuring the L2TP Peer Resynchronization Method

The L2TP implementation on MX Series routers supports resynchronization between a failed L2TP endpoint and its peer nonfailed endpoint. Peer resynchronization enables L2TP to recover from a daemon or router restart or a Routing Engine switchover.

L2TP peer resynchronization:

- Prevents the nonfailed endpoint from prematurely terminating a tunnel while the failed endpoint is recovering.
- Reestablishes the sequence numbers required for the operation of the L2TP control protocol.
- Resolves inconsistencies in the tunnel and session databases of the failed endpoint and the nonfailed endpoint.

You can configure the peer resynchronization method you want the router to use. Both the L2TP failover protocol method and the L2TP silent failover method are supported.

In Junos OS Releases through 15.1R5, 16.1R4, 16.2R1, and 17.1R1, the default behavior is for L2TP on the LAC to attempt to negotiate the L2TP failover protocol with the LNS. When the LNS supports this method and negotiation is successful, the L2TP failover protocol is used when either peer fails. When negotiation for L2TP failover protocol fails, then the peers use silent failover when either peer fails. This behavior is called `failover-protocol-fall-back-to-silent-failover`. Falling back to the silent failover method when failover protocol negotiation is unsuccessful prevents a subsequent peer failure from forcing a disconnection of the tunnel to the peer and all the associated sessions.

NOTE: The behavior just described applies when both peers are MX Series routers. If one endpoint is a third-party device, then the behavior for that device depends on its L2TP implementation.

You can disable the default behavior and force the LAC or the LNS to operate only in silent failover mode. This configuration can be useful when the peer routers either are configured for silent failover or incorrectly negotiate to use the failover protocol even though they do not support it. Another reason to use this statement is that the failover protocol method keeps the tunnel open with the failed peer, in case the failed peer is able to recover from the failure and resynchronize with the nonfailed peer. This behavior keeps the tunnel up and the subscribers logged in while traffic is not flowing, preventing service level agreements from being met. When you issue this statement and the peer supports only failover protocol, the nonfailed endpoint (LAC or LNS) uses silent failover for recovery.

To disable negotiation of the L2TP failover protocol:

- Configure disabling.

```
[edit services l2tp]
user@host# set disable-failover-protocol
```

Starting in Junos OS Releases 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, the default failover resynchronization method is changed to silent failover. Consequently, the `disable-failover-protocol` statement no longer needs to be used and is deprecated. If you upgrade from a lower-numbered release where the default method is `failover-protocol-fall-back-to-silent-failover`, and your configuration includes the `disable-failover-protocol` statement, the configuration is still supported, but the CLI notifies you that the statement is deprecated.

In these releases, you can still configure which method you want an endpoint to use, failover protocol or silent failover.

To configure the LAC or LNS to negotiate the L2TP failover protocol:

- Specify the failover protocol.

```
[edit services l2tp tunnel]
user@host# set failover-resync failover-protocol
```

If the negotiation fails, the endpoint falls back to the silent failover method.

To restore the default resynchronization method for the LAC or LNS:

- Specify the silent failover method.

```
[edit services l2tp tunnel]
user@host# set failover-resync silent-failover
```

Release History Table

Release	Description
15.1R6	Starting in Junos OS Release 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, silent failover is the default resynchronization method in Junos OS.
15.1R6	Starting in Junos OS Release 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, the <code>disable-failover-protocol</code> statement is deprecated, because the change in default resynchronization method makes it unnecessary.
15.1R6	Starting in Junos OS Releases 15.1R6, 16.1R5, 16.2R2, 17.1R2, and 17.2R1, the default failover resynchronization method is changed to silent failover. Consequently, the <code>disable-failover-protocol</code> statement no longer needs to be used and is deprecated.
15.1R5	In Junos OS Releases through 15.1R5, 16.1R4, 16.2R1, and 17.1R1, the default behavior is for L2TP on the LAC to attempt to negotiate the L2TP failover protocol with the LNS.

RELATED DOCUMENTATION

| [L2TP for Subscriber Access Overview](#) | 144

Tracing L2TP Events for Troubleshooting

IN THIS SECTION

- [Configuring the L2TP Trace Log Filename](#) | 327
- [Configuring the Number and Size of L2TP Log Files](#) | 328
- [Configuring Access to the L2TP Log File](#) | 328
- [Configuring a Regular Expression for L2TP Messages to Be Logged](#) | 329

- [Configuring Subscriber Filtering for L2TP Trace Operations | 329](#)
- [Configuring the L2TP Tracing Flags | 331](#)
- [Configuring the Severity Level to Filter Which L2TP Messages Are Logged | 331](#)

The Junos OS trace feature tracks L2TP operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

NOTE: This topic refers to tracing L2TP operations on MX Series routers. To trace L2TP operations on M Series routers, see [Tracing L2TP Operations](#).

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `j12tpd`. You can specify a different filename, but you cannot change the directory in which trace files are located.
2. When the trace log file *filename* reaches 128 kilobytes (KB), it is compressed and renamed *filename.0.gz*. Subsequent events are logged in a new file called *filename*, until it reaches capacity again. At this point, *filename.0.gz* is renamed *filename.1.gz* and *filename* is compressed and renamed *filename.0.gz*. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the [System Log Explorer](#).)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

The following topics describe how to configure all aspects of tracing L2TP operations:

Configuring the L2TP Trace Log Filename

By default, the name of the file that records trace output for L2TP is `j12tpd`. You can specify a different name with the `file` option.

To configure the filename for L2TP tracing operations:

- Specify the name of the file used for the trace output.

```
[edit services l2tp traceoptions]
user@host# set file l2tp_logfile_1
```

Configuring the Number and Size of L2TP Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format *.number.gz*. The newest archived file is *.0.gz* and the oldest archived file is *.(maximum number)-1.gz*. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, *filename*, reaches 2 MB, *filename* is compressed and renamed *filename.0.gz*, and a new file called *filename* is created. When the new *filename* reaches 2 MB, *filename.0.gz* is renamed *filename.1.gz* and *filename* is compressed and renamed *filename.0.gz*. This process repeats until there are 20 trace files. Then the oldest file, *filename.19.gz*, is simply overwritten when the next oldest file, *filename.18.gz* is compressed and renamed to *filename.19.gz*.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit services l2tp traceoptions]
user@host# set file l2tp_1_logfile_1 files 20 size 2097152
```

Configuring Access to the L2TP Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit services l2tp traceoptions]
user@host# set file l2tp_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit services l2tp traceoptions]
user@host# set file l2tp_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for L2TP Messages to Be Logged

By default, the trace operation output includes all lines relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit services l2tp traceoptions]
user@host# set file l2tp_1 _logfile_1 match regex
```

Configuring Subscriber Filtering for L2TP Trace Operations

Starting in Junos OS Release 14.1, you can apply filters to L2TP to limit tracing to particular subscribers or domains. Subscriber filtering simplifies troubleshooting in a scaled environment by enabling you to focus on a reduced set of trace results.

For subscriber usernames that have the expected form of *user@domain*, you can filter on the user, the domain, or both. You can use an asterisk (*) as a wildcard to substitute for characters at the beginning or end of either term or both terms to match a greater number of subscribers.

NOTE: You cannot filter results using a wildcard in the middle of the user or domain terms. For example, the following uses of the wildcard are not supported: tom*25@example.com, tom125@ex*.com.

When you enable filtering by username, traces that have insufficient information to determine the username are automatically excluded.

To configure subscriber filtering:

- Specify the filter.

```
[edit services l2tp traceoptions]
user@host# set filter user user@domain
```

NOTE: This syntax is different than the syntax used to filter subscribers on M Series routers.

Consider the following examples of using the wildcard for filtering:

- Filter results for the specific subscriber with the username, tom@example.com.

```
[edit services l2tp traceoptions]
user@host# set filter user tom@example.com
```

- Filter results for all subscribers whose username begins with tom.

```
[edit services l2tp traceoptions]
user@host# set filter user tom*
```

- Filter results for all subscribers whose username ends with tom.

```
[edit services l2tp traceoptions]
user@host# set filter user *tom
```

- Filter results for subscribers with the username tom at all domains beginning with ex.

```
[edit services l2tp traceoptions]
user@host# set filter user tom@ex*
```

- Filter results for all subscribers at all domains that end with ample.com.

```
[edit services l2tp traceoptions]
user@host# set filter user *ample.com
```

- Filter results for all subscribers whose username begins with tom at domains that end with example.com.

```
[edit services l2tp traceoptions]
user@host# set filter user tom*@example.com
```

Configuring the L2TP Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit services l2tp traceoptions]
user@host# set flag flag
```

Configuring the Severity Level to Filter Which L2TP Messages Are Logged

The messages associated with a logged event are categorized according to severity level. You can use the severity level to determine which messages are logged for the event type. A low severity level is less

restrictive—filters out fewer messages—than a higher level. When you configure a severity level, all messages at that level and all higher (more restrictive) levels are logged.

The following list presents severity levels in order from lowest (least restrictive) to highest (most restrictive). This order also represents the significance of the messages; for example, error messages are of greater concern than info messages.

- verbose
- info
- notice
- warning
- error

The severity level that you configure depends on the issue that you are trying to resolve. In some cases you might be interested in seeing all messages relevant to the logged event, so you specify `all`. You can also specify `verbose` with the same result, because `verbose` is the lowest (least restrictive) severity level; it has nothing to do with the terseness or verbosity of the messages. Either choice generates a large amount of output. You can specify a more restrictive severity level, such as `notice` or `info` to filter the messages. By default, the trace operation output includes only messages with a severity level of `error`.

To configure the type of messages to be logged:

- Configure the message severity level.

```
[edit services l2tp traceoptions]
user@host# set level severity
```

Release History Table

Release	Description
14.1	Starting in Junos OS Release 14.1, you can apply filters to L2TP to limit tracing to particular subscribers or domains. Subscriber filtering simplifies troubleshooting in a scaled environment by enabling you to focus on a reduced set of trace results.

RELATED DOCUMENTATION

| [L2TP for Subscriber Access Overview](#) | 144

5

CHAPTER

Configuring MPLS Pseudowire Subscriber Logical Interfaces

[MPLS Pseudowire Subscriber Logical Interfaces](#) | 334

MPLS Pseudowire Subscriber Logical Interfaces

IN THIS SECTION

- [Pseudowire Subscriber Logical Interfaces Overview | 334](#)
- [Anchor Redundancy Pseudowire Subscriber Logical Interfaces Overview | 338](#)
- [Configuring a Pseudowire Subscriber Logical Interface | 345](#)
- [Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router | 347](#)
- [Configuring a Pseudowire Subscriber Logical Interface Device | 348](#)
- [Changing the Anchor Point for a Pseudowire Subscriber Logical Interface Device | 350](#)
- [Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface | 353](#)
- [Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces | 354](#)
- [Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces | 355](#)
- [Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface | 357](#)
- [Configuring a PWHT with VC 11 Type Support | 359](#)
- [Configuring Load Balancing Support for Subscriber Traffic | 362](#)

Pseudowire Subscriber Logical Interfaces Overview

Subscriber management supports the creation of subscriber interfaces over point-to-point MPLS pseudowires. The pseudowire subscriber interface capability enables service providers to extend an MPLS domain from the access-aggregation network to the service edge, where subscriber management is performed. Service providers can take advantage of MPLS capabilities such as failover, rerouting, and uniform MPLS label provisioning, while using a single pseudowire to service a large number of DHCP and PPPoE subscribers in the service network.

NOTE: Pseudowire subscriber logical interfaces are supported on Modular Port Concentrators (MPCs) with Ethernet Modular Interface Cards (MICs) only. PPPoE and L2TP termination is not supported when VPLS encapsulation and DHCP authentication is used for the transport logical interface. However, Broadband Subscriber Management Layer 2 Wholesale functionality is supported with VPLS encapsulation. A dynamic VLAN interface is created with VPLS encapsulation on a wholesaler router, that performs VLAN tag switching to terminate PPPoE/

DHCP subscribers on the retailer network. For details, see [Broadband Subscriber Management Layer 2 Wholesale Topology and Configuration Elements](#).

The pseudowire is a tunnel that is either an MPLS-based Layer 2 VPN or Layer 2 circuit. The pseudowire tunnel transports Ethernet encapsulated traffic from an access node (for example, a DSLAM or other aggregation device) to the MX Series router that hosts the subscriber management services. The termination of the pseudowire tunnel on the MX Series router is similar to a physical Ethernet termination, and is the point at which subscriber management functions are performed. A service provider can configure multiple pseudowires on a per-DSLAM basis and then provision support for a large number of subscribers on a specific pseudowire.

Figure 19 on page 335 shows an MPLS network that provides subscriber management support.

At the access node end of the pseudowire, the subscriber traffic can be groomed into the pseudowire in a variety of ways, limited only by the number and types of interfaces that can be stacked on the pseudowire. You specify an anchor point, which identifies the logical tunnel interface that terminates the pseudowire tunnel at the access node.

Figure 19: MPLS Access Network with Subscriber Management Support

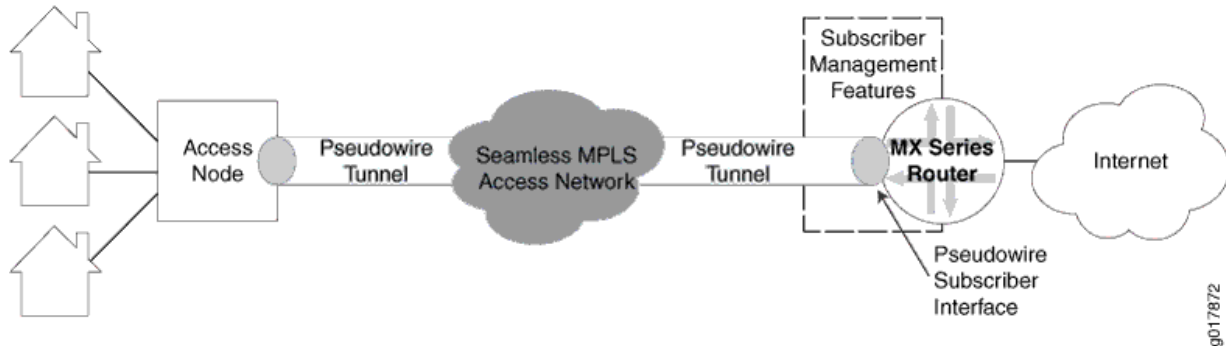


Figure 20 on page 337 shows the protocol stack for a pseudowire subscriber *logical interface*. The pseudowire is a virtual device that is stacked above the logical tunnel anchor point on the physical interface (the IFD), and supports a circuit-oriented Layer 2 protocol (either Layer 2 VPN or Layer 2 circuit). The Layer 2 protocol provides the transport and service logical interfaces, and supports the protocol family (IPv4, IPv6, or PPPoE).

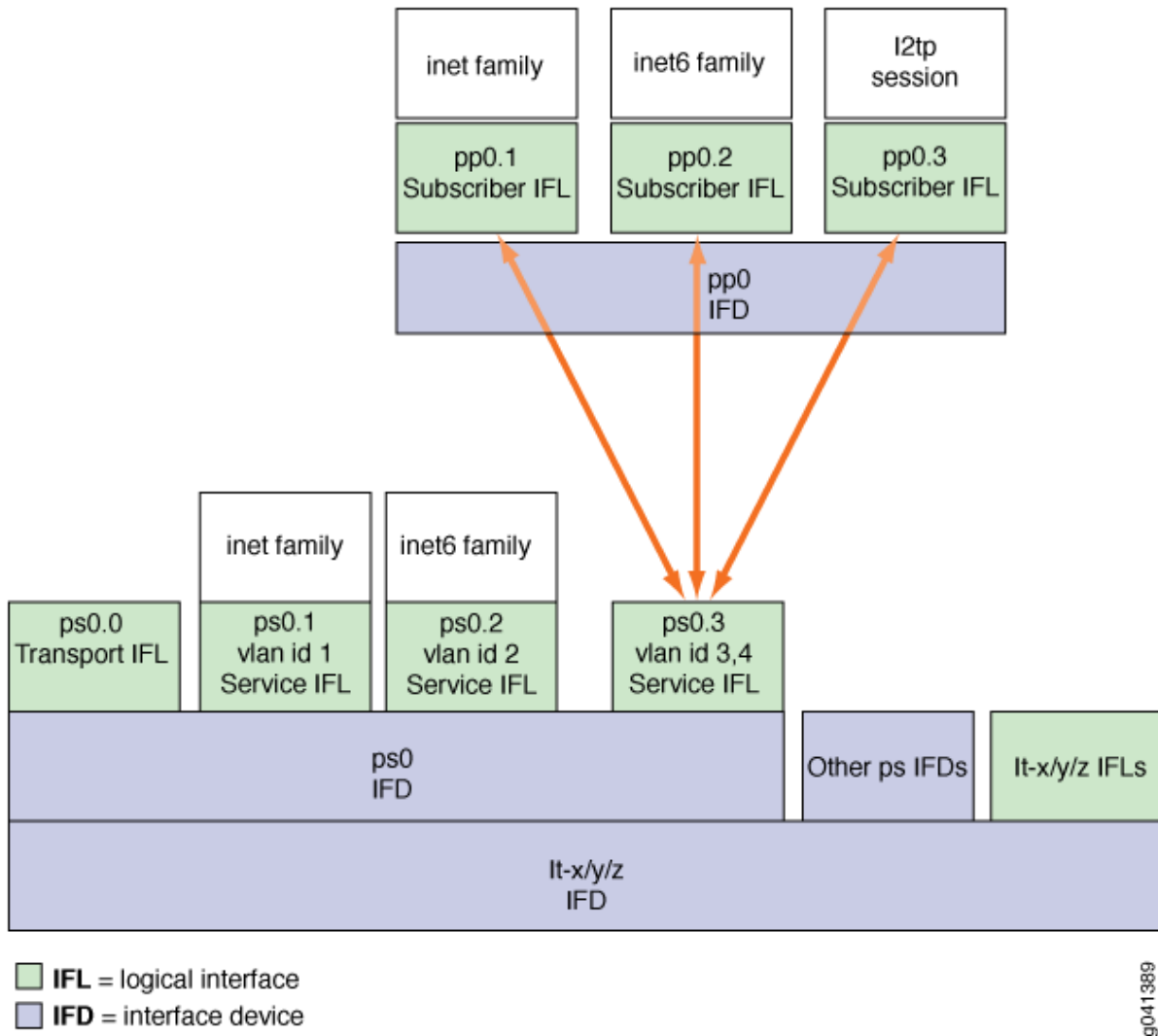
Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the support for pseudowire subscriber service interface over redundant logical tunnels is introduced in Layer 3 VPNs and draft-rosen multicast VPNs. Earlier, Layer 3 VPNs provided support for pseudowire subscriber services over logical tunnel interfaces only, and these interfaces used unicast routing protocols, such as OSPF or BGP. With this support, you can provision a multicast routing protocol, Protocol Independent Multicast (PIM), on the pseudowire subscriber interfaces, which gets terminated on the virtual routing

and forwarding (VRF) routing instance. Additionally, there is an increase in the scaling numbers of the pseudowire logical interface devices that provides additional resiliency support for pseudowire subscriber interfaces on redundant logical tunnel interfaces.

NOTE: When a pseudowire subscriber service interface is anchored to a redundant logical tunnel whose member interface (or FPC) does not exist, the tunnel interface comes down. In such cases, the pseudowire interfaces (physical and logical) should also be down, but however, the pseudowire subscriber logical interface state remains up, although the Layer 2 circuit services, such as ping toward a customer edge (CE) device from the service side of the pseudowire subscriber service interface, are not available.

This is because the transport side of the pseudowire subscriber logical interface stays up causing the services to be up.

Figure 20: Pseudowire Subscriber Interface Protocol Stack



The pseudowire configuration is transparent to the subscriber management applications and has no impact on the packet payloads that are used for subscriber management. Subscriber applications such as DHCP and PPPoE can be stacked over Layer 2 similar to the way in which they are stacked over a physical interface.

Starting with Junos OS release 16.1R1, family inet and family inet6 are supported on the services side of an MPLS pseudowire subscriber as well as non-subscriber logical interface.

Starting with Junos OS Release 16.1R1, Inline IPFIX is supported on the services side of an MPLS pseudowire subscriber logical interface.

Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, CCC encapsulation is supported on the transport side of an MPLS pseudowire subscriber logical interface.

Prior to Junos OS Release 19.1R1, the only supported encapsulation type on the pseudowire subscriber interfaces included:

- Transport logical interfaces—Circuit cross-connect (CCC) encapsulation.
- **Service logical interfaces:**
 - Ethernet VPLS encapsulation
 - VLAN bridge encapsulation
 - VLAN VPLS encapsulation

Starting in Junos OS Release 19.1R1, additional encapsulations are added to the pseudowire subscriber transport and service logical interfaces. The transport logical interface supports Ethernet VPLS encapsulation, and provisions for terminating the interface on the `l2backhaul-vpn` routing-instance. The service logical interface supports circuit cross-connect (CCC) encapsulation, and provisions for terminating the interface on locally switched Layer 2 circuits.

With the support of additional encapsulation types, you can benefit from demux of a `l2backhaul` VPN into multiple VPN services, such as Layer 2 circuit and Layer 3 VPN. Because pseudowire subscriber interfaces are anchored on redundant logical tunnels, this enhancement also provides line card redundancy.

Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, distributed denial-of-service (DDoS) protection is supported on the services side of an MPLS pseudowire subscriber logical interface.

Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, Policer and Filter are supported on the services side of an MPLS pseudowire subscriber logical interface.

Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, accurate transmit statistics on logical interface are supported on the services side of an MPLS pseudowire subscriber logical interface.

Starting with Junos OS Release 17.3R1 and later releases, stateful anchor point redundancy support is provided for pseudowire subscriber logical interface by the underlying redundant logical tunnel interface (rt) in active-backup mode. This redundancy protects the access and the core facing link against anchor PFE (Packet Forwarding Engine) failure.

Anchor Redundancy Pseudowire Subscriber Logical Interfaces Overview

In MPLS pseudowire deployments that use pseudowire subscriber logical interfaces, failure of the Packet Forwarding Engine hosting the logical tunnel that anchors those logical interfaces leads to traffic loss and subsequent subscriber session loss.

The Packet Forwarding Engine does not rely on the control plane for failure detection; instead it uses a liveness detection mechanism, with an underlying heartbeat-based algorithm, to detect the failure of other Packet Forwarding Engines in the system. The failure of a Packet Forwarding Engine also indicates the failure of the hosted logical tunnel, which ultimately lead to session loss. To avoid this session loss, a redundant anchor point is required to which the session can be moved without losing any traffic.

Starting from Junos OS Release 17.3 onward, pseudowire subscriber logical interfaces can be instantiated over an underlying redundant logical tunnel (rlt) interface in active-backup mode. This is in addition to installing pseudowires over a single logical tunnel interfaces. The most noticeable advantage of implementing the pseudowire subscriber logical interface over redundant logical tunnel interfaces is to provide redundancy of the underlying forwarding path.

Prior to Junos OS Release 18.3R1, you could specify a maximum of 2048 pseudowire subscriber redundant logical tunnel interface devices for an MX Series router. Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the pseudowire redundant logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.

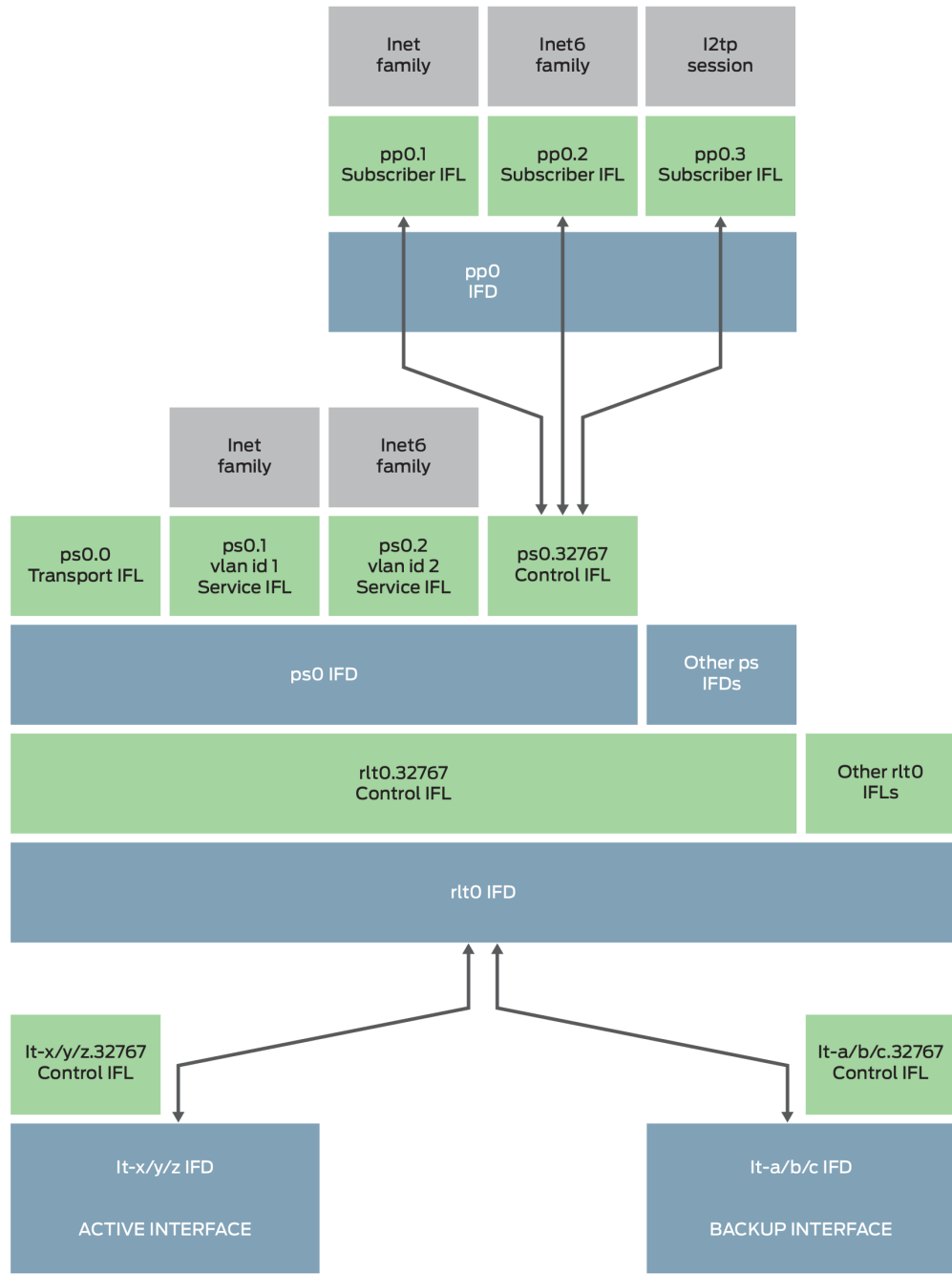
Junos OS Release 17.3 also supports an enhanced aggregated infrastructure for a Packet Forwarding Engine to provide anchor point redundancy. Enhanced aggregated infrastructure requires a minimum of one control logical interface that needs to be created on a redundant logical tunnel interface. Both transport and services logical interfaces created for the pseudowire subscriber logical interface are stacked on the underlying control logical interface for the redundant logical tunnel. This stacking model is used for both redundant and nonredundant pseudowire subscriber logical interfaces.

The following events have to trigger the removal of the physical interface from a redundant group:

- Hardware failure on Modular PIC Concentrator (MPC) or Modular Interfaces Card (MIC).
- MPC failure due to microkernel crash.
- MPC or MIC taken offline administratively.
- Power failure on an MPC or a MIC.

[Figure 21 on page 340](#) provides the details of pseudowire subscriber logical interface stacking over a redundant logical tunnel interface.

Figure 21: Pseudowire Subscriber Logical Interface Stacking over Redundant Logical Tunnel Interface



NOTE: Static service ifl is not stacked over transport ifl when RLT is used.

8043624

By default, [Link Protection](#) for redundant tunnel interfaces is revertive. In case of the active link failure, traffic is routed through the backup link. When the active link is reestablished, traffic is automatically routed back to the active link. This causes traffic loss and subscriber session loss.

To overcome the traffic and session loss, you can configure nonrevertive link protection for redundant tunnel interfaces by using the configuration statement `set interfaces rltX logical-tunnel-options link-protection non-revertive`. With this configuration, when the active link is reestablished, traffic is not routed back to the active link and continue to be forwarded on the backup link. Therefore, there is no traffic loss or subscriber session loss. You can also manually switch traffic from the backup link to the active link by using the `request interface (revert | switchover) interface-name` command.



CAUTION: The manual switching of the traffic incurs traffic loss.

NOTE:

- A control logical interface is created implicitly on an redundant tunnel interface with the pseudowire subscriber logical interface configuration and thus no additional configuration is needed.
- A redundant logical tunnel interface allows 32 member logical tunnel physical interfaces. However, a pseudowire subscriber logical interface hosted on the redundant logical tunnel interface limits the number of logical tunnel physical interfaces to two.

NOTE: You cannot disable the underlying redundant logical tunnel (rlt) interface or the underlying logical tunnel (lt) interface when a pseudowire is anchored on that interface. If you want to disable the underlying interface, you must first deactivate the pseudowire.

Starting in Junos OS Release 18.4R1, the support for inline distribution of single-hop Bidirectional Forwarding Detection (BFD) sessions is extended to pseudowire subscriber over redundant logical tunnel interfaces. For pseudowire subscriber over logical tunnel interfaces, the interfaces are anchored on a single Flexible PIC Concentrator (FPC), as a result, the inline distribution of single-hop BFD sessions is supported by default. With pseudowire redundant logical interfaces, the member logical tunnel interfaces can be hosted on different linecards. Because the distribution address is not available for the redundant logical interfaces, the distribution of single-hop BFD sessions was operated in a centralized mode before Junos OS Release 18.4R1.

With the support for inline distribution of single-hop BFD sessions over pseudowire redundant logical interfaces, there is a scaling advantage of up to 2000 single-hop BFD sessions at an interval of one second, and improvement in detection time enhancing the performance of the sessions.

The BFD operation for pseudowire subscriber over redundant logical interfaces is as follows:

1. When a new BFD session gets added it can either be anchored on an active or a backup FPC.
2. When either of the FPCs fail or reboot, all the sessions hosted on that FPC go down, and re-anchoring is triggered for the next available distribution address. The BFD sessions come back up after the sessions are installed on the other FPC and BFD packet exchange is started.

However, it is also possible that the sessions on the backup FPC might not go down when active FPC fails depending on the BFD detection time configured, as the forwarding state for the new active FPC might take some time to be programmed.

3. When the active FPC fails, all the BFD sessions get anchored on the backup FPC. Similarly, if the backup FPC fails, all the BFD sessions get anchored on the active FPC.
4. The BFD session re-anchoring is not triggered when the active FPC is online again.
5. With the non-revertive behavior enabled, when the previously active FPC is online again, the sessions are not expected to go down. With the default revertive behavior, it is possible that forwarding state needs to be updated and depending on the detection time configuration, the session may or may not flap.

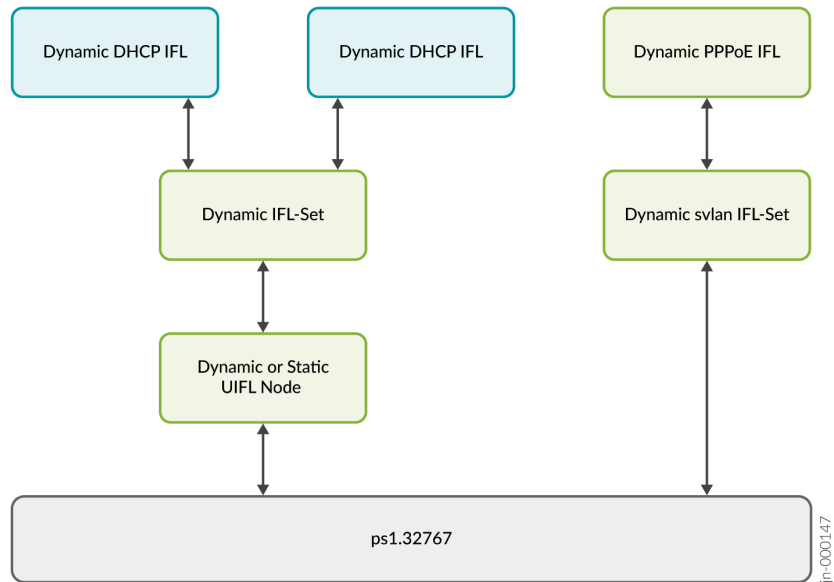
NOTE: Take the following into consideration with the support of inline distribution of single-hop BFD sessions on pseudowire subscriber over logical tunnel interfaces:

- On FPC type MPC 7e, with the activation of 7000 routing instance, it takes about six minutes for the 7000 BGP sessions to get established on the pseudowire subscriber interfaces anchored on redundant logical tunnel interfaces.
- A new system log error message - JTASK_SCHED_SLIP - is recorded during nonstop active routing (NSR). This is expected behavior of NSR with high scale and can be safely ignored, unless there are other issues, such as session flaps, that require action to be taken.

Starting in Junos OS Release 21.4R1, we've introduced CoS support for a BNG on subscriber-interface on pseudowire over an active-active redundant logical tunnel (RLT) interface for subscriber applications such as DHCP and PPPoE. This CoS property is achieved by providing the scheduling nodes for the logical tunnel links. For dynamic interfaces, interface sets, static underlying interfaces, and dynamic underlying interfaces over RLT, CoS allocates scheduling nodes for each link in the RLT, which has multiple logical tunnel links in active-active mode. In case of targeted interfaces and targeted interface sets, which have primary and backup links, CoS allocates scheduling nodes on the primary and backup links to optimize the use of scheduling nodes. Traffic for the subscriber targeted interfaces will be distributed to all the primary LT links when CoS is applied at the subscriber level. Also, traffic from any given subscriber is always processed by the same Packet Forwarding Engine.

Figure 22 on page 343 provides the details of the parent and child interfaces used for the four-level scheduler hierarchy for subscriber access. The dynamic PPPoE IFL and dynamic IFL-set are child nodes. The dynamic svlan IFL-set and dynamic or static uifl node are parent nodes.

Figure 22: Four-level Scheduler Hierarchy for Subscriber Access



When you enable targeting in a node, you must enable targeting for all the child nodes for CoS to function properly. To enable the child nodes, configure the dynamic profile at the [edit interfaces ps1 auto-configure stacked-vlan-ranges dynamic-profile]. Create dynamic profile by configuring dynamic targeted interfaces and interface sets at the [edit dynamic-profiles].

Here's an example of the dynamic profile configuration:

```

dvlanProf {
  interfaces {
    "$junos-interface-ifd-name" {
      unit "$junos-interface-unit" {
        demux-source [ inet inet6 ];
        no-traps;
        proxy-arp;
        vlan-tags outer "$junos-stacked-vlan-id" inner "$junos-vlan-id";
        targeted-distribution;
        family inet {
          unnumbered-address 100.0.0.1 preferred-source-address 100.0.0.1;
        }
      }
    }
  }
}
  
```

```

    }
    family inet6 {
        unnumbered-address lo0.0 preferred-source-address 1000:0::1;
    }
    family pppoe {
        duplicate-protection;
        dynamic-profile pppoeClientSvlanSetVar;
    }
}
}
}
}
}

```

```

pppoeClientSvlanSetVar {
    interfaces {
        interface-set "$junos-svlan-interface-set-name" {
            targeted-distribution;
            interface pp0 {
                unit "$junos-interface-unit";
            }
        }
        pp0 {
            unit "$junos-interface-unit" {
                actual-transit-statistics;
                ppp-options {
                    pap;
                }
                pppoe-options {
                    underlying-interface "$junos-underlying-interface";
                    server;
                }
                targeted-distribution;
                keepalives interval 30;
                family inet {
                    unnumbered-address "$junos-loopback-interface";
                }
            }
        }
    }
}
}
}
}

```

Also, you must configure the network-services enhanced-ip at the [edit chassis] hierarchy level because this feature works only in enhanced IP mode.

The active-active multiple link mode with targeting uses the targeting algorithms for RLT interface to distribute clients among the different RLT member (primary/secondary leg pairs). Targeting can be applied for dynamic subscribers and dynamic interface sets. The targeting algorithm goes through the list of pseudo IFLs associated with the member link pair and selects the first pseudo IFL that has sufficient capacity based on the configured rebalance-subscriber-granularity.

When targeting is enabled, the subscriber is assigned a default targeting weight based on the client type. The targeting algorithm uses allocation weight in the pseudo IFL selection process and IFL's debit weight is the weight counted against the assigned pseudo IFL. For all objects except the IFLset, the allocation and debit weight are the same and you can modify through the client profile. In case of the IFLset, only the allocation weight attribute can be modified through the client profile, and debit weight for the IFLset is fixed at a value of 0.

Table 29: Default Weights for Different Client Types

Client Type	Allocation Weight	Debit Weight
Dvlan	1	1
IpDemux	1	1
PPP	1	1
IFLset	32	0

Configuring a Pseudowire Subscriber Logical Interface

A pseudowire subscriber logical interface terminates an MPLS pseudowire tunnel from an access node to the MX Series router that hosts subscriber management, and enables you to perform subscriber management services at the interface.

To create a pseudowire subscriber logical interface:

1. Specify the number of pseudowire logical interfaces that the router can support.

See ["Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router"](#) on page 347.

2. Configure the pseudowire subscriber logical interface device.
See ["Configuring a Pseudowire Subscriber Logical Interface Device"](#) on page 348.
3. Configure the transport logical interface.
See ["Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface"](#) on page 353.
4. Configure the signaling for the pseudowire subscriber interface. You can use either Layer 2 circuit signaling or Layer 2 VPN signaling. The two signaling types are mutually exclusive for a given pseudowire.
 - To configure Layer 2 circuit signaling, see ["Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces"](#) on page 354.
 - To configure Layer 2 VPN signaling, see ["Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces"](#) on page 355.
5. Configure the service logical interface.
See ["Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface"](#) on page 357.
6. Configure the underlying interface device.
See *Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces*.
7. Configure CoS parameters and BA classification.
See [CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces](#).
8. (Optional) Associate a dynamic profile with the pseudowire subscriber logical interface.
You can associate DHCP, PPPoE, IP demux, and VLAN dynamic profiles with pseudowire subscriber logical interfaces. The support is similar to the typical Ethernet interface support.

NOTE: When using a PPPoE dynamic profile to create a pseudowire subscriber logical interface over a demux interface device, the dynamic profile must explicitly specify the correct pseudowire interface device over which the interface is created. The dynamic profile does not automatically create the interface over the demux0 interface device, as is the case with a VLAN demux interface.

9. (Optional) Configure interface set support for pseudowire subscriber logical interfaces.
See [Configuring Interface Sets](#) and [Understanding Interface Sets](#).
10. (Optional) Stack PPPoE logical interfaces over a pseudowire logical device.
11. (Optional) Load balancing support for subscriber traffic on pseudowire service (PS) interface. See [Configuring load balancing support for subscriber traffic](#).

Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router

You must set the maximum number of pseudowire logical interface devices (pseudowire tunnels) that the router can use for subscriber logical interfaces. Setting the maximum number also defines the interface names for the pseudowire interfaces. When you subsequently configure the interfaces, you must specify the interface names in the range from ps0 up to ps(*device-count - 1*).

For example, if you set the maximum number of devices to 5, then you can configure only interfaces ps0, ps1, ps2, ps3, and ps4.

Before Junos OS Release 17.2R1, you could specify a maximum of 2048 pseudowire logical interface devices for an MX Series router. Starting in Junos OS Release 17.2R1, on MX Series routers with MPC and MIC interfaces, the pseudowire logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.

Similarly, before Junos OS Release 18.3R1, you could specify a maximum of 2048 pseudowire subscriber redundant logical tunnel (rlt) interface devices for an MX Series router. Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the pseudowire redundant logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.

Starting in Junos OS Release 20.4R1, on MX2010 and MX2020 routers with the MX2K-MPC9E or MX2K-MPC11E line card, you can specify up to 18000 pseudowire logical interface devices.

The PFE hosting the maximum pseudowire logical interface devices provides the configuration flexibility needed for special cases that might occur for business edge scenarios. However, you can exceed the available PFE resources as you configure additional services on the pseudowire logical interface devices ports. To support a scaled configuration, ensure that you populate the appropriate number of PFEs for the chassis, and that you distribute the pseudowire logical interface devices across the PFEs in such a way that ensures that no PFE is overwhelmed by the anticipated peak load. As part of the network planning for your particular deployment, you must consider the exact mix of the distribution of the pseudowire logical interface devices and the services associated with the devices.

BEST PRACTICE: A configured pseudowire logical interface device consumes resources from shared pools even when the device has no active subscriber logical interfaces. To conserve resources, do not deploy an excessive number of pseudowire devices that you do not intend to use.

To configure the number of pseudowire logical interface devices that you want the router to support:

1. Specify that you want to configure the pseudowire service.

```
[edit chassis]
user@host# edit pseudowire-service
```

2. Set the maximum number of pseudowire logical interface devices.

```
[edit chassis pseudowire-service]
user@host# set device-count 500
```

Configuring a Pseudowire Subscriber Logical Interface Device

To configure a pseudowire logical interface device that the router uses for subscriber logical interfaces, you specify the logical tunnel that processes the pseudowire termination. You can also use redundant logical tunnels to provide redundancy for member logical tunnels. You can configure additional optional parameters for the interface device, such as VLAN tagging method, MTU, and gratuitous ARP support.

NOTE: You must create a logical tunnel for the pseudowire logical interface device. If you are using redundant logical tunnels, you must create the redundant tunnel.

To configure the pseudowire subscriber interface device:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

NOTE: The available interface names are determined by the `[edit chassis pseudowire-service device-count]` statement. The names you specify must be in the range `ps0` through `ps(device-count - 1)`. If you specify an interface name outside that range, the pseudowire interface is not created.

```
user@host# edit interfaces ps0
```

2. Specify the logical tunnel interface that is the anchor point for the pseudowire logical interface device. The anchor point must be an `lt` device in the format `lt-fpc/pic/port`.



CAUTION: Do not reconfigure the logical tunnel interface that is associated with the pseudowire subscriber interface device unless you first deactivate all subscribers that are using the pseudowire subscriber interface.

NOTE: Tunnel services must be enabled on the `lt` interface that is the anchor point or a member link in a redundant logical tunnel. You use the command, `set chassis fpc slot-number pic pic-number tunnel-services bandwidth bandwidth` to enable tunnel services.

NOTE: You cannot disable the underlying logical tunnel (`lt`) interface or redundant logical tunnel (`rlt`) interface when a pseudowire is anchored on that interface. If you want to disable the underlying interface, you must first deactivate the pseudowire.

```
[edit interfaces ps0]
user@host# set anchor-point lt-1/0/10
```

3. (Optional) Specify the MAC address for the pseudowire logical interface device.

NOTE: You should ensure that you change the MAC address before passing traffic or binding subscribers on the pseudowire port. Changing the MAC address when the pseudowire port is active (for example, while an upper layer protocol is negotiating) can negatively impact network performance until adjacencies learn of the new MAC address.

```
[edit interfaces ps0]
user@host# set mac 00:00:5E:00:53:55
```

4. (Optional) Specify the VLAN tagging method used for the pseudowire logical interface device. You can specify single tagging, dual (stacked) tagging, mixed (flexible) tagging, or no tagging.

```
[edit interfaces ps0]
user@host# set flexible-vlan-tagging
```

See [Enabling VLAN Tagging](#) for additional information about VLAN tagging.

5. (Optional) Specify the encapsulation type for the pseudowire logical interface device.

Starting in Junos OS Release 19.1R1, you can configure additional encapsulations – Ethernet VPLS and circuit cross-connect-based encapsulations – for the transport and service pseudowire subscriber logical interface devices, respectively.

```
[edit interfaces]
user@host# set logical-interface-unit encapsulation encapsulation-type
```

6. (Optional) Specify the MTU for the pseudowire logical interface device. If you do not explicitly configure the MTU, the router uses the default value of 1500.

```
[edit interfaces ps0]
user@host# set mtu 2500
```

See [Setting the Protocol MTU](#) for additional information.

7. (Optional) Specify that the pseudowire logical interface device does not respond to gratuitous ARP requests.

```
[edit interfaces ps0]
user@host# set no-gratuitous-arp-request
```

See [Configuring Gratuitous ARP](#) for additional information.

8. (Optional) Specify that reverse-path forwarding checks are performed for traffic on the pseudowire logical interface device.

```
[edit interfaces ps0]
user@host# set rpf-check
```

See [Understanding Unicast RPF \(Routers\)](#) for additional information.

9. Configure additional optional parameters for the pseudowire logical interface device, such as [description](#), *apply-groups*, *apply-groups-except*, and traceoptions.

Changing the Anchor Point for a Pseudowire Subscriber Logical Interface Device

You cannot dynamically change an anchor point that has active pseudowire devices stacked above it. You must commit certain changes before you move the anchor point. Examples of this situation include

moving the anchor point from one logical tunnel to another logical tunnel, from a logical tunnel to a redundant logical tunnel, and from a redundant logical tunnel to a logical tunnel.

To move the anchor point between logical tunnel interfaces:

1. Deactivate the stacked pseudowires and commit. This may require bringing down any subscribers using the pseudowires.

```
[edit interfaces]
user@host# deactivate psnumber
user@host# commit
```

2. Change the anchor on the deactivated pseudowire to the new logical tunnel interface and commit.

```
[edit interfaces]
user@host# set psnumber anchor-point lt-fpc/pic/port
user@host# commit
```

3. Reactivate the stacked pseudowires and commit.

```
[edit interfaces]
user@host# activate psnumber
user@host# commit
```

To move the anchor point from a logical tunnel interface to a redundant logical tunnel interface:

1. Deactivate the stacked pseudowires and commit. This may require bringing down any subscribers using the pseudowires.

```
[edit interfaces]
user@host# deactivate psnumber
user@host# commit
```

2. Add the new redundant logical tunnel interface and commit.

- a. Create the tunnel and set the maximum number of devices allowed.

```
[edit chassis]
user@host# set redundancy-group interface-type redundant-logical-tunnel device-count count
```

- b. Bind each member logical tunnel to the redundant logical tunnel.

NOTE: Redundant logical tunnels require members to be in active-backup mode. The backup logical tunnel must be on a different FPC than the active logical tunnel. For example, if the active tunnel is on FPC 3, then the backup tunnel must be on a different FPC, such as FPC 4.

```
[edit interfaces rltnumber]
user@host# set redundancy-group member-interface lt-fpc/pic/port active
user@host# set redundancy-group member-interface lt-fpc/pic/port backup
```

- c. Commit your changes.

```
[edit interfaces rltnumber]
user@host# commit
```

3. Change the anchor on the deactivated pseudowire to the new redundant logical tunnel interface and commit.

```
[edit interfaces]
user@host# set psnumber anchor-point rltnumber
user@host# commit
```

4. Reactivate the stacked pseudowires and commit.

```
[edit interfaces]
user@host# activate psnumber
user@host# commit
```

To move the anchor point from a redundant logical tunnel interface to a logical tunnel interface that is a member of the redundant logical tunnel:

1. Deactivate the stacked pseudowires; this may require bringing down any subscribers using the pseudowires. Delete the redundant logical tunnel interface and commit your changes.

```
[edit interfaces]
user@host# deactivate psnumber
```

```
user@host# delete rlnumber
user@host# commit
```

2. Change the anchor on the deactivated pseudowire to the new logical tunnel interface and commit.

```
[edit interfaces]
user@host# set psnumber anchor-point lt-fpc/pic/port
user@host# commit
```

3. Reactivate the stacked pseudowires and commit.

```
[edit interfaces]
user@host# activate psnumber
user@host# commit
```

Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface

This topic describes how to configure a pseudowire transport logical interface. A pseudowire device can have only one transport logical interface.

A pseudowire logical device and its related pseudowire logical interfaces are dependent on the state of the underlying logical transport interface device, which is either the Layer 2 VPN or Layer 2 circuit.

NOTE: We recommend that you use unit 0 to represent the transport logical interface for the pseudowire device. Non-zero unit numbers represent *service* logical interfaces used for pseudowire subscriber interfaces.

To configure a pseudowire transport logical interface:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

```
[edit]
user@host# edit interfaces ps0
```

2. Specify that you want to configure unit 0, which represents the transport logical interface.

```
[edit interfaces ps0]
user@host# edit unit 0
```

3. (Optional) Specify the encapsulation method for the transport logical interface.

Starting in Junos OS Release 19.1R1, you can configure Ethernet VPLS encapsulation, in addition to circuit cross-connect-based encapsulations for pseudowire subscriber transport logical interfaces.

```
[edit interfaces ps0 unit 0]
user@host# set encapsulation ethernet-ccc
user@host# set encapsulation ethernet-vpls
user@host# set family vpls core-facing
```

4. (Optional) Configure the termination of the transport logical interface on l2backhaul-vpn routing-instance. This support is enabled from Junos OS Release 19.1R1.

```
[edit routing-instances routing-instance-name]
user@host# set vlan-model one-to-one instance-role access instance-type l2backhaul-vpn
interface ps1.0s
user@host# set no-local-switching
```

Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces

This topic describes the steps for configuring Layer 2 circuit signaling used for the pseudowire subscriber logical interface support. You can also use Layer 2 VPN signaling for pseudowire subscriber logical interfaces. The two methods are mutually exclusive; you can use only one method for a particular pseudowire.

To configure Layer 2 circuit signaling for pseudowire interfaces:

1. Specify that you want to configure Layer 2 circuit parameters at the protocols hierarchy level.

```
[edit protocols]
user@host# edit l2circuit
```

2. Specify the IP address of the neighbor, to identify the PE router used for the Layer 2 circuit.

```
[edit protocols l2circuit]
user@host# edit neighbor 192.168.102.15
```

3. Specify the interface used by the Layer 2 circuit traffic.

```
[edit protocols l2circuit neighbor 192.168.102.15]
user@host# edit interface ps1.0
```

4. Configure the virtual circuit ID that identifies the Layer 2 circuit for the pseudowire.

```
[edit protocols l2circuit neighbor 192.168.102.15 interface ps1.0]
user@host# set virtual-circuit-id 5
```

For more information about Layer 2 circuits, see *Configuring Interfaces for Layer 2 Circuits*.

Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces

This topic describes the steps for configuring Layer 2 VPN signaling used for the pseudowire subscriber logical interface support. You can also use Layer 2 circuit signaling for pseudowire subscriber logical interfaces. The two methods are mutually exclusive; you can use only one method on a particular pseudowire.

To configure Layer 2 VPN signaling for pseudowire interfaces:

1. Specify the name of the routing instance you want to configure.

```
[edit]
user@host# edit routing-instances l2vpn0
```

2. Configure the Layer 2 VPN routing instance type.

```
[edit routing-instances l2vpn0]
user@host# set instance-type l2vpn
```

3. Associate the pseudowire logical interface for the Layer 2 VPN.

```
[edit routing-instances l2vpn0]
user@host# set interface ps1.0
```

4. Configure the unique identifier for the routes that belong to the Layer 2 VPN.

```
[edit routing-instances l2vpn0]
user@host# set route-distinguisher 198.51.100.101100
```

5. Configure the VPN routing and forwarding (VRF) target of the routing instance.

```
[edit routing-instances l2vpn0]
user@host# set vrf-target target:10:100
```

6. Specify that you want to configure the Layer 2 VPN protocol for the routing instance.

```
[edit routing-instances l2vpn0]
user@host# edit protocols l2vpn
```

7. Configure the encapsulation type for the routing instance.

```
[edit routing-instances l2vpn0 protocols l2vpn]
user@host# set encapsulation-type ethernet
```

8. Specify the site name and site identifier for the Layer 2 VPN.

```
[edit routing-instances l2vpn0 protocols l2vpn]
user@host# set site PE1 site-identifier 1
```

9. Specify the interface that connects to the site, and the remote interface to which you want the specified interface to connect.

```
[edit routing-instances l2vpn0 protocols l2vpn]
user@host# set interface ps1.0 remote-site-id 2
```


10. Configure the tracing options for traffic that uses the Layer 2 VPN.

```
[edit routing-instances l2vpn0 protocols l2vpn]
user@host# set traceoptions file l2vpn flag all
```

Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface

This topic describes how to configure a pseudowire service logical interface. Service logical interfaces represent the attachment circuits for pseudowire logical interfaces.

As described in the "[Pseudowire Subscriber Logical Interfaces Overview](#)" on page 334, you can choose whether to configure a service logical interface together with a higher subscriber logical interface, depending upon the business need. In a broadband edge configuration, the higher subscriber logical interface is the demarcation point for subscribers. However, in a business edge configuration, the service logical interface is the demarcation point for the business subscribers, and also serves as the subscriber logical interface, so no subscriber logical interfaces are explicitly configured.

NOTE: Non-zero unit numbers represent *service* logical interfaces used for pseudowire subscriber interfaces. Use unit 0 to represent the *transport* logical interface for the pseudowire device.

To configure a pseudowire service logical interface:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

```
[edit]
user@host# edit interfaces ps0
```

2. Configure the unit for the service logical interface. Use a non-zero unit number.

```
[edit interfaces ps0]
user@host# edit unit 1
```

3. (Optional) Specify the encapsulation type for the service logical interface.

Starting in Junos OS Release 19.1R1, you can configure circuit cross-connect-based encapsulations, in addition to the Ethernet VPLS, VLAN bridge, and VLAN VPLS encapsulations for pseudowire subscriber service logical interfaces.

The pseudowire subscriber service logical interfaces support single-tagged traffic, double-tagged traffic, and list of VLANs on the single logical interface.

```
[edit interfaces ps0]
user@host# set unit 1 encapsulation vlan-ccc
user@host# set vlan-id vlan-ID
user@host# set vlan-tags outer outer-tag inner inner-tag
user@host# set vlan-id-list vlan-id-list
user@host# set family ccc
```

4. (Optional) Configure filters and policers on the family circuit cross-connect encapsulation.

```
[edit interfaces ps0]
user@host# set unit 1 family ccc filter group
user@host# set unit 1 family ccc filter input input-list
user@host# set unit 1 family ccc filter output output-list
user@host# set unit 1 family ccc policer input
user@host# set unit 1 family ccc policer output
```

5. Configure the VLAN tag IDs.

```
[edit interfaces ps0 unit 1]
user@host# set vlan-tags outer 1 inner 1
```

6. Configure the interface to respond to ARP requests when the device has an active route to the ARP request target address.

```
[edit interfaces ps0 unit 1]
user@host# set proxy-arp
```

7. Specify that you want to configure the protocol family information. Pseudowire service logical interfaces support IPv4 (inet), IPv6 (inet6), and PPPoE (pppoe) protocol families.

For example, to configure the IPv4 family:

- a. Specify that you want to configure IPv4.

```
[edit interfaces ps0 unit 1]
user@host# edit family inet
```

- b. Configure the parameters for the family.

```
[edit interfaces ps0 unit 1 family inet]
user@host# set filter input filter 1 output filter 4
user@host# set mac-validate loose
user@host# set input-hierarchical-policer policer-1
user@host# set unnumbered-address lo0.0 preferred-source-address 198.51.100.11
```

8. (Optional) Configure the termination of the service logical interface on locally switched Layer 2 circuits. This support is enabled from Junos OS Release 19.1R1.

```
[edit protocols]
user@host# set l2circuit local-switching interface ps0.1 encapsulation-type ethernet-vlan
ignore-encapsulation-mismatch ignore-mtu-mismatch
```

Configuring a PWHT with VC 11 Type Support

You can configure a pseudowire headend termination (PWHT) interface on a service PE router and configure ethernet-tcc encapsulation on the pseudowire subscriber (PS) transport logical interface. When you use this feature, the service PE router does not have to support TDM/SONET/SDH-encapsulated traffic coming from access-side customers. The IP-based point-to-point pseudowire—which is an LDP-signaled FEC 128 (virtual circuit (VC) type 11)—connects the service PE router to the access device that is connected to the CE router. You configure the pseudowire to terminate into a Layer 3 VPN instance or a global IP table.

The feature supports IPv4 and IPv6 payloads and unicast and multicast traffic.

The service PE router uses ARP mediation to resolve Layer 2 addresses when different resolution protocols are used on either end of a circuit. To the service PE router, the access CE router appears as though locally connected. This ARP mediation is provided by proxy ARP on IPv4 addresses and by Neighbor Discovery Protocol (NDP) on IPv6 addresses. The service PE router creates a local ARP entry that corresponds to the access CE router's IPv4 address or adds the access CE router's IPv6 address to the neighbor table.

Before you configure the interfaces and the `l2circuit` protocol for the PWHT with VC 11 type support:

- Configure the target LDP session for the Layer 2 circuit. See *Configuring LDP for Layer 2 Circuits*.
- Configure the Layer 3 VPN. See *Introduction to Configuring Layer 3 VPNs*.

NOTE: When you enable `family tcc` and `encapsulation ethernet-tcc` on a PS interface, note the following constraints on the configuration:

- Support for only one IP pseudowire per PS physical interface
- No support for a control word; for BFD over the PS interface; or for active-standby, hot-standby, or all-active configuration on the IP pseudowire

To configure PWHT on the service PE router with termination into a Layer 3 VPN instance:

1. Configure the redundant logical tunnel (RLT) with this command:

```
set chassis redundancy-group interface-type redundant-logical-tunnel device-count;
```

2. Configure the interfaces—Configure the redundancy group and member interfaces on the `rlt` interface; configure the anchor point, which is on the `rlt` interface; and configure the PS transport and service logical interfaces. Configure `family tcc` and `encapsulation-type ethernet-tcc` on the transport logical interface. See an example of the interfaces configuration just after the Note.

NOTE:

- Configure only one PS service logical interface.
- ARP could be generated on the service PE router for all IP addresses within the subnet configured on the PS service logical interface. To prevent generation of many ARPs, we recommend that you use a /30 or /31 subnet on the PS service logical interface.

```
edit interfaces {
  rlt0 {
    redundancy-group {
      member-interface lt-1/0/10;
      member-interface lt-2/0/10;
    }
  }
  ps1 {
```

```

    anchor-point {
        rlt0;
    }
    unit 0 {                                     >>> PS transport IFL
+     family tcc;
+     encapsulation ethernet-tcc;
    }
    unit 1 {                                     >>> PS service IFL
        family inet {
            address 10.9.1.1/30;
        }
        family inet6 {
            address 2001:db8:600::9189/126;
        }
    }
}

```

3. Configure the `l2circuit` protocol and include the `send-ip-addr-list-tlv` statement to signal that an IP TLV is sent. Configure the encapsulation type on the transport logical interface as `internetworking`. Here's an example of the protocol configuration:

```

[edit protocols]
  l2circuit {
    neighbor 10.10.255.1 {
      interface ps1.0 {
+       send-ip-addr-list-tlv;
        virtual-circuit-id 100;
        encapsulation-type interworking;
      }
    }
  }
}

```

You can use the following `show` commands to view the results of this configuration:

- Use the `show route table l2circuit.0` command to see that VC type 11 has been enabled.
- Use the `show l2circuit connections extensive` command to see that encapsulation is set to `internetworking`.
- Use the `show route table mpls.0 protocol l2circuit` command to see that the label route and tcc route for forwarding the traffic out of the IP pseudowire and into the IP pseudowire have been added.

Configuring Load Balancing Support for Subscriber Traffic

Configure the RLT with the router's LT links in active-active mode. RLT applications can be enhanced to include LT child member links as an aggregated property.

Starting in Junos OS Release 21.4R1, we provide load balancing support for subscriber sessions on the PS interface over multiple LT child member links of the RLT at the same time. The load balancing property of the RLT interface allows subscriber traffic on the PS interface to be dispersed and load-balanced over different PICs and line-cards.

For RLT interface supports PS anchor point redundancy to enhance LAG mode. Use the `enhanced-ip` option or the `enhanced-ethernet` option at the `[edit chassis network-services]` hierarchy level while configuring PS IFD anchored on RLT.

Computed hash is used in selecting an ECMP path and load balancing. You can configure load balancing for IPv4 traffic over Layer 2 Ethernet pseudowires. You can also configure load balancing for Ethernet pseudowires based on IP information.

Limitations

- The BNG load balancing support on the pseudowire subscriber (PS) interface feature is supported only for all trio-based line-cards supporting the BBE access model on the MX Series routers.
- You cannot change the PS anchor point unless you disable the PS physical interface.
- Transient traffic disruption may occur when you add or remove an RLT member. Adding or removing RLT member link behaviour is similar to any other aggregate interface behaviour.
- Ingress stats for each LT member are not available. However, aggregate PS IFL or IFD statistics are available for both directions.
- RLT active-active mode is supported only for subscriber services.

Below are not supported for the current load-balancing support on PS over RLT over multiple active child LT links

- PS over RLT interface support on MX240, MX480, and MX960 line cards.
- CoS support of hierarchical policer interface for active-active mode member links
- CoS aggregated Ethernet support for subscriber traffic on pseudowire service (PS) interface
- L2 Service IFL and business-edge (L3) support for active-active mode member link
- PS interface support on non redundant
- Hierarchical CoS support for anchor point redundancy of pseudowire subscriber logical Interfaces

To configure the load balancing support for subscriber traffic:

1. Configure the extended DHCP local server options on the router, see [Configure a Router as an Extended DHCP Local Server](#).
2. Configure two logical tunnels on two different line cards to create a redundant logical tunnel (RLT).

```
[edit]
user@host# set chassis redundancy graceful-switchover
user@host# set chassis aggregated-devices ethernet device-count number
user@host# set chassis pseudowire-service device-count number
user@host# set chassis redundancy-group interface-type redundant-logical-tunnel device-count
number
user@host# set chassis fpc 1 pic 2 tunnel-services bandwidth value
user@host# set chassis fpc 2 pic 0 tunnel-services bandwidth value
user@host# set chassis fpc 3 pic 3 tunnel-services bandwidth value
user@host# set chassis network-services enhanced-ip
```

3. Configure RLT interface and include the logical tunnel interface in the redundancy group by configuring member-interface interface-name. Configuring RLT interface, see [Configuring a Pseudowire Subscriber Logical Interface Device](#)
4. Configure dynamic profiles for subscriber management, see [Dynamic Profiles for Subscriber Management](#).
5. Configuring l2circuit with a backup neighbor that has the same virtual-circuit-id, see [Example: Configuring Longest Match for LDP](#).
6. Tunnel egress bandwidth utilisation can be verified using LT interface egress statistics. View your configuration for PS over RLT active-active mode support.

```
apply-groups [ global re0 ];
system {
    commit synchronize;
    configuration-database {

    }
chassis {
    aggregated-devices {
        ethernet {
            device-count 4;
        }
    }
    pseudowire-service {
```

```
        device-count 100;
    }
    redundancy-group {
        interface-type {
            redundant-logical-tunnel {
                device-count 1;
            }
        }
    }
}
fpc 1 {
    pic 1 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}
fpc 2 {
    pic 1 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}
network-services enhanced-ip;
}
access-profile none;
interfaces {
    inactive: traceoptions {
        file dcd.log size 100m;
        flag all;
    }
    ge-1/1/1 {
        mtu 1522;
        unit 0 {
            family inet {
                address 172.10.0.1/30;
            }
            family iso;
            family mpls;
        }
    }
    ge-1/1/2 {
        flexible-vlan-tagging;
```



```
unit 0 {
    vlan-id 1;
    family inet {
        address 200.0.0.1/24;
    }
    family inet6 {
        address 200::1/64;
    }
}
}
fxp0 {
    unit 0 {
        family inet;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.2/32;
            address 100.0.0.1/32 {
                primary;
            }
            address 101.0.0.1/32;
        }
        family iso {
            address 49.0001.0192.0168.0002.00;
        }
        family inet6 {
            address 1000::1/128;
        }
    }
}
}
ps0 {
    anchor-point {
        rlt0;
    }
    flexible-vlan-tagging;
    auto-configure {
        vlan-ranges {
            dynamic-profile vlan-prof-1 {
                accept any;
                ranges {
                    1-400;
                }
            }
        }
    }
}
```

```
        }
    }
}
remove-when-no-subscribers;
}
encapsulation flexible-ethernet-services;
mac 22:22:22:22:22:24;
no-gratuitous-arp-request;
unit 0 {
    encapsulation ethernet-ccc;
}
}
r1t0 {
    logical-tunnel-options {
        link-protection {
            non-revertive;
        }
    }
    redundancy-group {
        member-interface lt-1/1/10 {
            active;
        }
        member-interface lt-2/1/10 {
            active;
        }
    }
    unit 0 {
        encapsulation ethernet;
        peer-unit 1;
    }
    unit 1 {
        encapsulation ethernet;
        peer-unit 0;
    }
}
}
policy-options {
    prefix-list dhcp-voip;
    policy-statement export-dhcp {
        from protocol access-internal;
        then accept;
    }
}
}
```

```
.....  
.....  
routing-options {  
    router-id 192.168.0.2;  
    autonomous-system 65000;  
}  
protocols {  
    router-advertisement {  
        interface all;  
    }  
    neighbor-discovery;  
    ppp-service {  
        inactive: traceoptions {  
            file size 500m files 5;  
            level all;  
            flag all;  
        }  
    }  
    bgp {  
        group evpn {  
            local-address 192.168.0.2;  
            family evpn {  
                signaling;  
            }  
            peer-as 65000;  
            local-as 65000;  
            neighbor 192.168.0.1;  
        }  
    }  
    isis {  
        interface ge-1/1/1.0;  
        interface fxp0.0 {  
            disable;  
        }  
        interface lo0.0 {  
            passive;  
        }  
        export export-dhcp;  
    }  
    l2circuit {  
        neighbor 192.168.0.1 {  
            interface ps0.0 {  
                virtual-circuit-id 1000010200;  
            }  
        }  
    }  
}
```

```

        encapsulation-type ethernet;
        ignore-mtu-mismatch;
        no-vlan-id-validate;
        pseudowire-status-tlv;
    }}}
    ldp {
        interface ge-1/1/1.0;
        interface lo0.0;
    }
    mpls {
        interface ge-1/1/1.0;
        interface fxp0.0 {
            disable;
        }
    }
    pppoe {
        inactive: traceoptions {
            file pppoe.log size 500m files 5;
            level all;
            flag all;
        }
    }

```

SEE ALSO

show interfaces terse

show interfaces statistics

show subscribers summary

show l2circuit connections

Release History Table

Release	Description
21.4R1	Starting in Junos OS Release 21.4R1, we've introduced CoS support for BNG on subscriber-interface on pseudowire (PS) over active-active redundant logical tunnel (RLT) interface for subscriber applications such as DHCP and PPPoE.
21.4R1	Starting in Junos OS Release 21.4R1, we provide load balancing support for subscriber sessions on the PS interface over multiple LT child member links of the RLT at the same time. The load balancing property of the RLT interface allows subscriber traffic on the PS interface to be dispersed and load-balanced over different PICs and line-cards.

21.2R1	Starting in Junos OS Release 21.2R1, you can configure a PWHT interface on a service PE router with ethernet-tcc encapsulation on the interface. The pseudowire is VC type 11.
20.4R1	Starting in Junos OS Release 20.4R1, on MX2010 and MX2020 routers with the MX2K-MPC9E or MX2K-MPC11E line card, you can specify up to 18000 pseudowire logical interface devices.
19.1R1	Starting in Junos OS Release 19.1R1, additional encapsulations are added to the pseudowire subscriber transport and service logical interfaces. The transport logical interface supports Ethernet VPLS encapsulation, and provisions for terminating the interface on the l2backhaul-vpn routing-instance. PPPoE and L2TP termination is not supported when VPLS encapsulation is used for the transport logical interface. The service logical interface supports circuit cross-connect (CCC) encapsulation, and provisions for terminating the interface on locally switched Layer 2 circuits.
19.1R1	Starting in Junos OS Release 19.1R1, you can configure additional encapsulations – Ethernet VPLS and circuit cross-connect-based encapsulations – for the transport and service pseudowire subscriber logical interface devices, respectively.
19.1R1	Starting in Junos OS Release 19.1R1, you can configure Ethernet VPLS encapsulation, in addition to circuit cross-connect-based encapsulations for pseudowire subscriber transport logical interfaces.
19.1R1	Starting in Junos OS Release 19.1R1, you can configure circuit cross-connect-based encapsulations, in addition to the Ethernet VPLS, VLAN bridge, and VLAN VPLS encapsulations for pseudowire subscriber service logical interfaces.
18.4R1	Starting in Junos OS Release 18.4R1, the support for inline distribution of single-hop Bidirectional Forwarding Detection (BFD) sessions is extended to pseudowire subscriber over redundant logical tunnel interfaces.
18.3R1	Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the support for pseudowire subscriber service interface over redundant logical tunnels is introduced in Layer 3 VPNs and draft-rosen multicast VPNs.
18.3R1	Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the pseudowire redundant logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.
18.3R1	Starting in Junos OS Release 18.3R1, on MX Series routers with MPC and MIC interfaces, the pseudowire redundant logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.

17.3R1	Starting with Junos OS Release 17.3R1 and later releases, stateful anchor point redundancy support is provided for pseudowire subscriber logical interface by the underlying redundant logical tunnel interface (rlt) in active-backup mode. This redundancy protects the access and the core facing link against anchor PFE (Packet Forwarding Engine) failure.
17.2R1	Starting in Junos OS Release 17.2R1, on MX Series routers with MPC and MIC interfaces, the pseudowire logical interface devices scaling numbers has increased to 7000 devices to provide additional resiliency support.
16.1R1	Starting with Junos OS release 16.1R1, family inet and family inet6 are supported on the services side of an MPLS pseudowire subscriber as well as non-subscriber logical interface.
16.1R1	Starting with Junos OS Release 16.1R1, Inline IPFIX is supported on the services side of an MPLS pseudowire subscriber logical interface.
15.1R3	Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, CCC encapsulation is supported on the transport side of an MPLS pseudowire subscriber logical interface.
15.1R3	Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, distributed denial-of-service (DDoS) protection is supported on the services side of an MPLS pseudowire subscriber logical interface.
15.1R3	Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, Policer and Filter are supported on the services side of an MPLS pseudowire subscriber logical interface.
15.1R3	Starting with Junos OS Release 15.1R3 and 16.1R1 and later releases, accurate transmit statistics on logical interface are supported on the services side of an MPLS pseudowire subscriber logical interface.

RELATED DOCUMENTATION

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview](#)

[CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces](#)

[Tunnel Interface Configuration on MX Series Routers Overview](#)

[Router Chassis Configuration Statements](#)



CHAPTER

Wi-Fi Access Gateways

Wi-Fi Access Gateways | 372

Wi-Fi Access Gateways

IN THIS SECTION

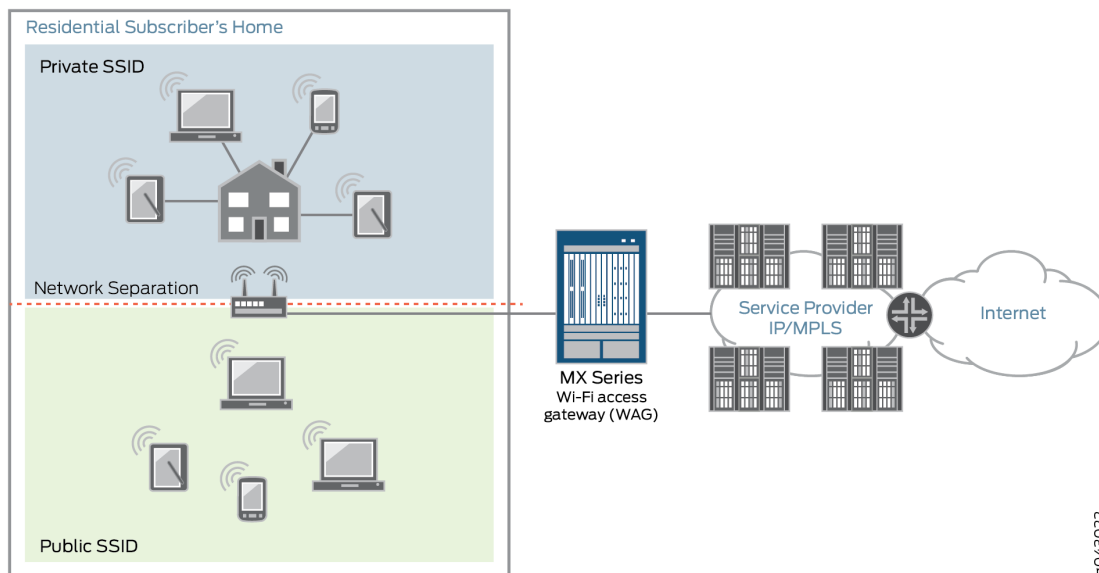
- [Wi-Fi Access Gateway Overview | 372](#)
- [Wi-Fi Access Gateway Deployment Model Overview | 374](#)
- [Supported Access Models for Dynamic-Bridged GRE Tunnels on the Wi-Fi Access Gateway | 376](#)
- [Wi-Fi Access Gateway Configuration Overview | 377](#)
- [Configuring a Pseudowire Subscriber Logical Interface Device for the Wi-Fi Access Gateway | 377](#)
- [Configuring Conditions for Enabling Dynamic-Bridged GRE Tunnel Creation | 379](#)
- [Configuring VLAN Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways | 382](#)
- [Configuring Untagged Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways | 387](#)

Wi-Fi Access Gateway Overview

Wi-Fi access gateway (WAG) provides the public with Wi-Fi access from a residential Wi-Fi network or from a business Wi-Fi network. At home, subscribers have their existing Wi-Fi network; however, a part of their network is available for the general public to use. Members of the public who have an account with the same Internet service provider as the subscriber has at home can access the Internet and mobile network through the public part of the subscriber's Wi-Fi connection when they are in close proximity to the subscriber's home. WAG authenticates and connects subscribers regardless of their physical location.

Starting in Junos OS Release 17.2R1, service providers can deploy the MX Series router as a broadband network gateway (BNG) within their network, and then deploy the BNG as a WAG. [Figure 23 on page 373](#) shows a sample topology.

Figure 23: MX Series Router Deployed as a WAG



After a WAG has been deployed, service providers can configure the WAG to create secure wireless home network connections for computers, laptops, and other Wi-Fi electronic products (such as game systems, tablets, or mobile phones). WAG offers wireline and mobile service providers the following deployments and business value opportunities:

- **Wireline service providers**—The WAG deployment is based on an in-home division of access points or public access points, and works with any Wi-Fi access point that creates a generic routing encapsulation (GRE) tunnel to the MX Series router. This deployment protects subscribers and reduces churn by including free Wi-Fi with a paid wireline subscription. For added value, service providers can also sell ad hoc access or mode, such as airport, public safety, search-and-rescue, and café access.
- **Mobile service providers**—The WAG deployment is based on the mobile service provider's own access points, or wholesale and retail with the wireline service provider. Service providers that offer *quadruple play*, where TV, Internet, wireless, and landline phone services are combined, can leverage both wireline and wireless assets. This deployment offsets costs in mobile packet core and radio access network infrastructures with the ability to offload mobile data. For added value, service providers can offer Wi-Fi for all devices with a mobile data plan as a competitive differentiator.

Customers who purchase broadband can also receive Wi-Fi on any community Wi-Fi access point. Subscribers have a private and secure home connection, and can also access a public connection that is shared by other subscribers. To maintain a level of security and protect the private home connection, the two networks are separated. This separation ensures a strong level of bandwidth on the subscribers' personal connections.

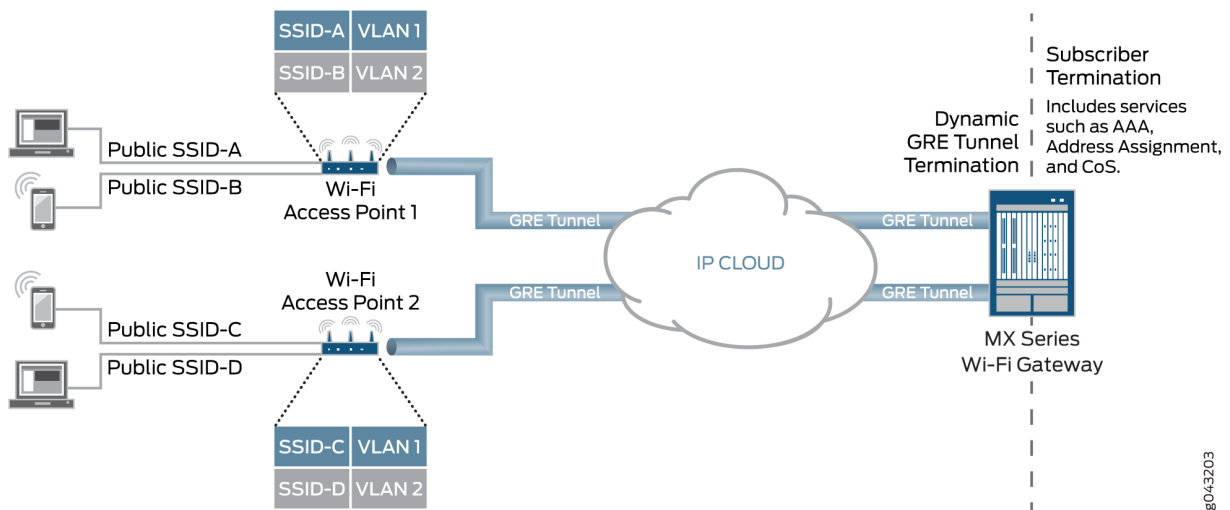
Subscriber services such as authentication, authorization, and accounting (AAA); address assignment; hierarchical quality of service (QoS); lawful intercept; and class of service (CoS) are supported for individual Dynamic Host Configuration Protocol (DHCP) subscribers within the GRE tunnels. Using GRE tunnels for Wi-Fi provides the following benefits:

- Wi-Fi users who are not directly connected through Layer 2 to WAG are authenticated because GRE tunnels transmit Layer 2 information across any IP network.
- Services based on user equipment-specific information are applied using the media access control (MAC) address or Subscriber Identity Module (SIM) card.
- Services are applied in the network, not just at the Wi-Fi access point.
- The soft GRE or Ethernet-over-GRE standard is supported on most Wi-Fi access points. For services using the Ethernet over GRE standard, only one side of the tunnel needs to be configured; the other end learns the remote IP addresses of all remote tunnel endpoints by examining the incoming GRE packets.

Wi-Fi Access Gateway Deployment Model Overview

[Figure 24 on page 375](#) shows an MX Series router broadband network gateway (BNG) deployed as a Wi-Fi access gateway (WAG). The WAG provides a multiservice edge with a full broadband feature set that is highly reliable because of the included redundant hardware. Ethernet frames from the user equipment device must be tunneled to the BNG across an IP cloud or public Internet.

Figure 24: MX Series as Wi-Fi Access Gateway Deployment Model



To support the MX Series BNG deployed as a WAG, dynamic-bridged generic routing encapsulation (GRE) tunnels are created and terminated at the BNG when it receives GRE traffic from the wireless access point (WAP). Dynamic Host Configuration Protocol (DHCP) subscribers are transported through GRE tunnels as either VLAN-tagged per service set identifier (SSID) or untagged. When the user equipment device connects to the SSID and begins to send traffic, the access point initiates a Layer 2 soft GRE or Ethernet-over-GRE connection to the MX Series BNG and the BNG dynamically builds the GRE tunnel. GRE tunnels are cleared after all of the subscribers within a GRE tunnel have logged out and a configurable timer has expired.

This deployment model supports a full set of services per user equipment device and per access point. Subscriber services such as authentication, authorization, and accounting (AAA); address assignment; hierarchical quality of service (QoS); lawful intercept; and class of service (CoS) are supported for individual DHCP subscribers within the GRE tunnels. No additional service cards are required for GRE or QoS because all features run inline on MPCs.

External RADIUS proxy supports Extensible Authentication Protocol (EAP) Subscriber Identity Module (SIM), Tunnelled Transport Layer Security (TTLS), and Authentication and Key Agreement (AKA) protocols. The External RADIUS proxy also integrates with HTTP redirect to the Web portal.

The MX Series as WAG deployment model also supports the wholesale of access point access to multiple retail service providers. This wholesaling allows the local breakout of traffic or Layer 3 handoff to retail service providers.

Supported Access Models for Dynamic-Bridged GRE Tunnels on the Wi-Fi Access Gateway

IN THIS SECTION

- [Dynamic VLAN-Tagged Subscribers | 376](#)
- [Untagged Subscribers | 377](#)

Dynamic-bridged generic routing encapsulation (GRE) tunnels and the Wi-Fi access gateway support interface stacks for VLAN-tagged and untagged subscribers. Subscriber features such as dynamic and service profiles for DHCP subscribers, lawful intercept, firewall filters, and change of authorization (CoA) are supported.

Scaling limitations of pseudowire subscriber interface devices (*psn* IFDs) require that multiple tunnels share the same *psn* IFD. The pseudowire is a virtual device that is stacked above the logical tunnel anchor point on the physical interface (the IFD).

NOTE: The *psn* IFD used to service dynamic GRE tunnel terminations cannot be simultaneously used to service MPLS pseudowire terminations.

Subscriber services and lawful intercept are supported only at the IP demultiplexing (demux) interface level.

NOTE: A GRE tunnel cannot have both untagged and tagged subscribers.

The tagged model and the untagged model are described in the following sections:

Dynamic VLAN-Tagged Subscribers

To make provisioning and troubleshooting easier for VLAN-tagged subscribers, use the same set of VLANs on all of the Wi-Fi access points. Doing this requires that the same pseudowire subscriber interface service logical interface (*psn* IFL) (associated with a VLAN ID) on a *psn* IFD represents multiple GRE tunnels.

A dynamic VLAN demux interface (*demux0.yyyyyyy*) is created for each VLAN tag and is stacked over the tunnel *psn* interface (*psn.xxxxxxx*). There can be multiple VLANs (single and dual-tagged) over the

same GRE tunnel. The subscribers' IP demux interfaces are then created over the VLAN demux interface.

Untagged Subscribers

Untagged DHCP subscribers can be created directly over the GRE tunnel. For each subscriber, an IP demux interface (demux0.yyyyyyy) is created and is stacked over the tunnel psn logical interface (psn.xxxxxxxx). There can be multiple subscribers over the same GRE tunnel.

Wi-Fi Access Gateway Configuration Overview

To configure the MX Series router as a Wi-Fi access gateway (WAG):

1. Configure a pseudowire subscriber logical interface device.
See ["Configuring a Pseudowire Subscriber Logical Interface Device for the Wi-Fi Access Gateway" on page 377](#).
2. Configure the conditions for enabling dynamic-bridged GRE tunnels.
See ["Configuring Conditions for Enabling Dynamic-Bridged GRE Tunnel Creation" on page 379](#).
3. Configure the type of dynamic-bridged GRE tunnel that carries subscriber traffic to the WAG:

NOTE: A GRE tunnel cannot have both untagged and tagged subscribers.

- If the subscriber traffic is VLAN-tagged, see ["Configuring VLAN Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways" on page 382](#).
- If the subscriber traffic is untagged, see ["Configuring Untagged Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways" on page 387](#).

Configuring a Pseudowire Subscriber Logical Interface Device for the Wi-Fi Access Gateway

Before you begin, you must create a logical tunnel interface:

- Configure the maximum number of pseudowire logical interfaces devices. See ["Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router" on page 347](#).
- Configure a tunnel interface. See [Tunnel Interface Configuration on MX Series Routers Overview](#).

To configure the pseudowire subscriber logical interface device on which the dynamic-bridged GRE tunnel is built on the MX Series router Wi-Fi access gateway:

1. Specify that you want to configure the pseudowire subscriber logical interface device.

```
user@host# edit interfaces psn
```

For example:

```
user@host# edit interfaces ps0
```

2. Specify the logical tunnel interface that is the anchor point for the pseudowire logical device interface.

```
[edit interfaces psn]  
user@host# set anchor-point lt-fpc/pic/port
```

For example:

```
[edit interfaces ps0]  
user@host# set anchor-point lt-0/0/0
```

3. Configure three-level hierarchical scheduling on the logical tunnel interface.

```
[edit interfaces lt-fpc/pic/port]  
user@host# set hierarchical-scheduler implicit-hierarchy
```

For example:

```
[edit interfaces lt-0/0/0]  
user@host# set hierarchical-scheduler implicit-hierarchy
```

4. Configure the mixed VLAN tagging method for the pseudowire logical interface device.

```
[edit interfaces psn]  
user@host# set flexible-vlan-tagging
```

NOTE: You must configure `flexible-vlan-tagging` even if only untagged subscriber packets are being transported on the dynamic-bridged GRE tunnel.

For example:

```
[edit interfaces ps0]
user@host# set flexible-vlan-tagging
```

5. Specify that you want to configure unit 0, which represents the transport logical interface.

```
[edit interfaces psn]
user@host# edit unit 0
```

For example:

```
[edit interfaces ps0]
user@host# edit unit 0
```

6. Specify the Ethernet CCC encapsulation method for the transport logical interface.

```
[edit interfaces psn unit 0]
user@host# set encapsulation ethernet-ccc
```

For example:

```
[edit interfaces ps0 unit 0]
user@host# set encapsulation ethernet-ccc
```

Configuring Conditions for Enabling Dynamic-Bridged GRE Tunnel Creation

Before you begin:

- Configure the pseudowire logical device on which to build the dynamic-bridged GRE tunnel. See ["Configuring a Pseudowire Subscriber Logical Interface Device for the Wi-Fi Access Gateway" on page 377](#).
- Configure interface lo0 with the source IP address of the GRE tunnels for the Wi-Fi access gateway (WAG). Use the IP address of the MX Series router that you want to receive the incoming GRE traffic. This address cannot be the primary or preferred address on lo0. See [Configuring a Loopback Interface](#).

To configure the conditions for enabling dynamic-bridged generic routing encapsulation (GRE) tunnel creation on the MX Series router WAG, you configure one or more GRE tunnel groups. Multiple GRE tunnel groups can have the same source-address or the same destination-networks value, but you cannot use a specific source-address *and* destination-networks combination in more than one GRE tunnel group.

To configure a GRE tunnel group:

1. Name the dynamic GRE tunnel group.

```
[edit services]
user@host# set soft-gre group-name
```

For example:

```
[edit services]
user@host# set soft-gre AP-Group1
```

2. Specify the source IP address of the GRE tunnels for the WAG. Use the IP address of the MX Series router that you configured to receive the incoming GRE traffic.

```
[edit services soft-gre group-name]
user@host# set source-address wag-ip-address
```

For example:

```
[edit services soft-gre AP-Group1]
user@host# set source-address 192.168.0.20
```


3. Specify the IP subnets from which GRE traffic can be processed.

```
[edit services soft-gre group-name]  
user@host# set destination-networks [prefix]
```

For example:

```
[edit services soft-gre AP-Group1]  
user@host# set destination-networks 192.0.2.0/24
```

4. Specify the pseudowire subscriber interface device (IFD) on which to build the dynamic-bridged GRE tunnels.

```
[edit services soft-gre group-name]  
user@host# set service-interface psn
```

For example:

```
[edit services soft-gre AP-Group1]  
user@host# set service-interface ps0
```

5. Specify the dynamic profile that configures the GRE tunnel.

```
[edit services soft-gre group-name]  
user@host# set dynamic-profile profile-name
```

For example:

```
[edit services soft-gre AP-Group1]  
user@host# set dynamic-profile tunnel_profile
```

6. (Optional) Configure the number of seconds that a GRE tunnel remains up after the last subscriber session on the tunnel has ended.

```
[edit services soft-gre group-name]  
user@host# set tunnel-idle-timeout seconds
```

The default tunnel-idle-timeout value is 120 seconds.

For example:

```
[edit services soft-gre AP-Group1]
user@host# set tunnel-idle-timeout 60
```

7. To configure another GRE tunnel group, repeat this procedure.

Configuring VLAN Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways

To configure subscriber interfaces for VLAN-tagged Dynamic Host Configuration Protocol (DHCP) subscribers on dynamic-bridged generic routing encapsulation (GRE) tunnels:

1. Name the dynamic profile. that creates the GRE tunnel

```
[edit]
user@host# set dynamic-profiles profile-name
```

For example:

```
[edit]
user@host# set dynamic-profiles tunnel_profile
```

2. Define the interface with the internal variable used by the router to match the interface name of the receiving interface.

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces $junos-interface-ifd-name
```

For example:

```
[edit dynamic-profiles tunnel_profile]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Define the unit with the internal variable.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name]
user@host# set unit $junos-interface-unit
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name]
user@host# set unit $junos-interface-unit
```

4. (Optional) Enable packet reassembly for fragmented GRE packets.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set reassemble-packets
```

5. Define the unit family type.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set family (inet | inet6)
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set family inet
```

6. Enable the local address for the interface to be derived from the loopback interface address.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit family (inet | inet6)]
user@host# set unnumbered-address lo0.0
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit family inet]
user@host# set unnumbered-address 100.0
```

7. Configure the router to respond to any ARP request.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set proxy-arp
```

8. Configure stacked VLAN processing:

a. Access the VLAN range configuration for stacked VLANs.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# edit auto-configure stacked-vlan-ranges
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# edit auto-configure stacked-vlan-ranges
```

b. Specify the dynamic profile that is used to create VLANs.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges]
user@host# edit dynamic-profile profile-name
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges]
user@host# edit dynamic-profile auto_svlan_demux
```

- c. Specify that the VLAN dynamic profile accepts any type of VLAN Ethernet packet.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges dynamic-profile profile-name]
user@host# set accept any
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges dynamic-profile auto_svlan_demux]
user@host# set accept any
```

- d. Specify the outer and inner stacked VLAN ranges that you want the dynamic profile to use.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges dynamic-profile profile-name]
user@host# set ranges low-tag-high-tag,low-tag-high-tag
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure stacked-vlan-ranges dynamic-profile auto_svlan_demux]
user@host# set ranges 1000-1100,1200-1300
```

9. Configure single-tagged VLAN processing:

- a. Access the VLAN range configuration for single VLANs.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# edit auto-configure vlan-ranges
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# edit auto-configure vlan-ranges
```

- b. Specify the dynamic profile used to create VLANs.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges]
user@host# edit dynamic-profile profile-name
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges]
user@host# edit dynamic-profile auto_vlan_demux
```

- c. Specify that the VLAN dynamic profile accepts any type of VLAN Ethernet packet.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges dynamic-profile profile-name]
user@host# set accept any
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges dynamic-profile auto_vlan_demux]
user@host# set accept any
```

- d. Specify the VLAN range that you want the dynamic profile to use.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges dynamic-profile profile-name]
user@host# set ranges low-tag-high-tag
```

For example:

```
[edit dynamic-profiles tunnel_profile interfaces $junos-interface-ifd-name unit $junos-
interface-unit auto-configure vlan-ranges dynamic-profile auto_vlan_demux]
user@host# set ranges 1-50
user@host# set ranges 101-150
```

Configuring Untagged Subscriber Interfaces for Dynamic-Bridged GRE Tunnels on Wi-Fi Access Gateways

To configure subscriber interfaces for untagged Dynamic Host Configuration Protocol (DHCP) subscribers on dynamic-bridged generic routing encapsulation (GRE) tunnels:

1. Name the dynamic profile that creates the GRE tunnel.

```
[edit]
user@host# set dynamic-profiles profile-name
```

For example:

```
[edit]
user@host# set dynamic-profiles tunnel_demux
```

2. Define the interface with the internal variable used by the router to match the interface name of the receiving interface.

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces $junos-interface-ifd-name
```

For example:

```
[edit dynamic-profiles tunnel_demux]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Define the unit with the internal variable.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name]
user@host# set unit $junos-interface-unit
```

For example:

```
[edit dynamic-profiles tunnel_demux interfaces $junos-interface-ifd-name]
user@host# set unit $junos-interface-unit
```

4. (Optional) Enable packet reassembly for fragmented GRE packets.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set reassemble-packets
```

5. Configure the variable for the underlying interface of the demux interfaces.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set demux-options underlying-interface $junos-underlying-interface
```

For example:

```
[edit dynamic-profiles tunnel_demux interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set demux-options underlying-interface $junos-underlying-interface
```

6. Define the unit family type.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set family (inet | inet6)
```

For example:

```
[edit dynamic-profiles tunnel_demux interfaces $junos-interface-ifd-name unit $junos-
interface-unit]
user@host# set family inet
```

Release History Table

Release	Description
17.2R1	Starting in Junos OS Release 17.2R1, service providers can deploy the MX Series router as a broadband network gateway (BNG) within their network, and then deploy the BNG as a WAG.

7

CHAPTER

Fixed Wireless Access Networks

[Fixed Wireless Access Networks | 390](#)

[Tracing Fixed Wireless Access Events for Troubleshooting | 407](#)

Fixed Wireless Access Networks

IN THIS SECTION

- [Fixed Wireless Access Network Overview | 390](#)
- [How to Configure Fixed Wireless Access | 401](#)
- [Verifying and Monitoring Fixed Wireless Access | 406](#)

Fixed Wireless Access Network Overview

IN THIS SECTION

- [References | 394](#)
- [3GPP Fixed Wireless Access Terminology | 395](#)
- [Benefits of Fixed Wireless Access | 401](#)

Service providers can manage subscribers over a wireless network to the home instead of having to run fiber to the building. The subscribers are in a fixed location, typically a residence, with customer premises equipment (CPE) that exchanges wireless radio signals with the provider network. The wireless network uses a fiber backhaul tower to handle traffic for the last miles from a hard-wired network to the subscribers. Multiple towers can relay traffic between each other to and from the fiber optic network. Starting in Junos OS Release 19.2R1, MX Series routers acting as BNGs can support subscribers in Third-Generation Partnership Project (3GPP) fixed wireless access networks, enabling the integration of fixed wireless subscribers with the wireline subscriber management backend.

You can find a helpful summary of related terminology in "[Fixed Wireless Access Network Overview](#)" on [page 390](#).

[Figure 25 on page 391](#) shows a representative topology for a fixed wireless access network.

Figure 25: Fixed Wireless Access Network Topology

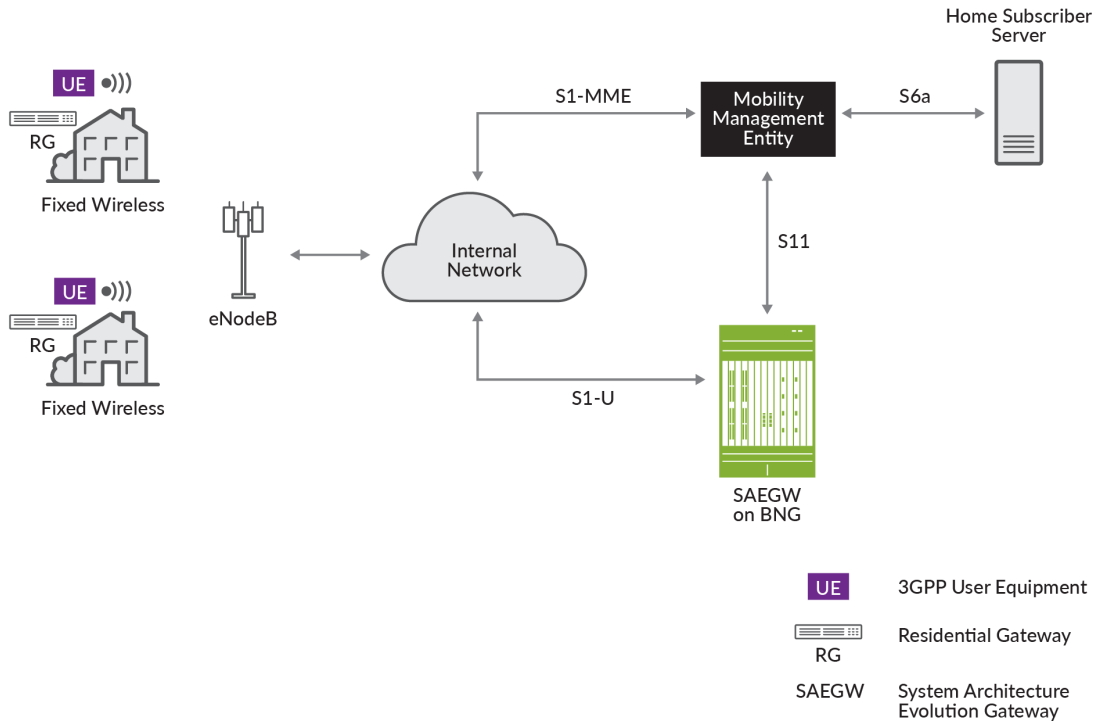
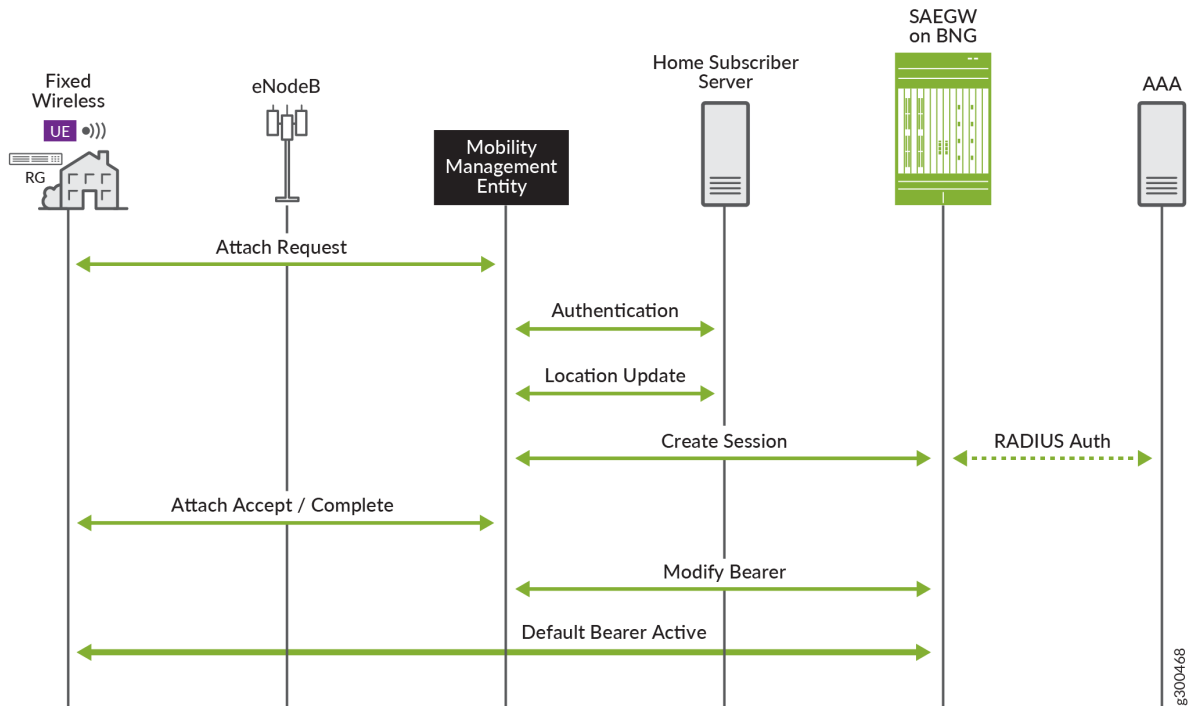


Figure 26 on page 392 shows the GPRS tunneling protocol (GTP) signaling messages in the fixed wireless access network that are necessary to activate and deactivate a default bearer. The messages are actually request and response pairs, but for simplicity the pairs are represented in the figure with bidirectional arrows. The Default Bearer Active line represents the activated default bearer that will carry data traffic.

NOTE: In this implementation, the UE roaming is not supported.

Figure 26: Signaling Messages for Bringing Default Bearer Up and Down



Briefly, the message exchange sequence is the following:

1. The user equipment (UE) sends an Attach Request to the eNodeB, which forwards the message to the mobility management entity (MME). The request includes the *APN*.
2. The MME exchanges Identity request/response messages with the UE to determine the International Mobile Subscriber Identity (*IMS*) or other identifier of the UE.
3. The MME sends the identifier to the Home Subscriber Server (HSS) to authenticate the UE.
4. The MME exchanges Location Update request/response messages with the HSS. In this exchange, the MME provides the HSS with its own address. The HSS acknowledges the update and sends subscription information for the UE from its database.
5. The MME sends a Create Session Request to the System Architecture Evolution gateway (SAEGW). The MME first compares the *APN* provided by the UE with the *APN* for which it is authorized according to the subscription information from the HSS. If there is a match, the MME includes that *APN* in the Create Session Request. If it does not match, then the MME instead includes the *APN* authorized by the HSS.
6. The SAEGW performs the following actions:
 - Validates information elements received in the request.

- Validates the APN requested by the subscriber. If you have configured authentication, the BNG communicates with the RADIUS server to accomplish that task.
 - Receives the R- *TEID-C*, which consists of the IP address of the S11 interface on the MME and the MME's allocated identifier.
 - Allocates the local TEIDs:
 - The L-TEID-C is the IP address of the S11 interface on the BNG and the BNG's allocated identifier.
 - The L-TEID-U is the IP address of the S1-U interface on the BNG and the BNG's allocated identifier.
 - Allocates an IP address for the pseudowire interface it creates for the session. The address comes from either a locally configured address pool on the BNG or from the RADIUS server.
 - Creates the session and sends the Create Session response to the MME. The response includes the IP address that the SAEGW has assigned for the bearer and both L-TEID-C and L-TEID-U.
7. The MME exchanges messages with the eNodeB, which then establishes the bearer component from the UE to eNodeB.
 8. The UE signals to the eNodeB that attachment is complete; the eNodeB notifies the MME.
 9. The MME and SAEGW exchange Modify Bearer request/response messages to determine the final parameters for the bearer.

When the BNG receives the request, it creates the dynamic pseudowire (ps) interface that will receive the GTP-U encapsulated data packets from the eNodeB. The BNG creates one dynamic ps interface per UE. The BNG also receives the R-TEID-U from the eNodeB in the Modify Bearer request. After creating the interface, the BNG sends a Modify Bearer response to the MME.

10. The default bearer is now active and subscriber data traffic can pass back and forth between the UE through eNodeB to the SAEGW and then the connected PDN.

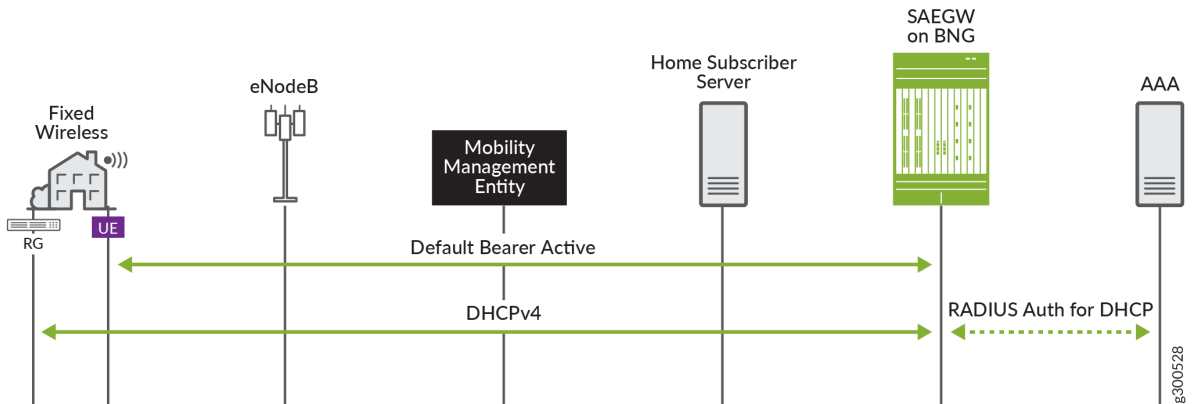
However, if the residential gateway is a DHCP client, it first begins the DHCP message exchange for the subscriber. The exchange takes place on the default bearer. After DHCP operations have completed to bind the subscriber and provide the DHCP configuration, then data traffic passes over the bearer.

The MME and SAEGW also exchange Delete Session request and response messages. When the SAEGW receives the request, it initiates the subscriber logout process, much as it would for a wireline DHCP subscriber.

[Figure 27 on page 394](#) shows the DHCP connection for the case where the RG creates DHCP subscribers. The DHCP control packets are exceptioned to the Routing Engine as for any other DHCP deployment. The behavior for creating and controlling the DHCP subscribers is the same as for a

wireline broadband network. When the subscriber is bound, the UE can then start sending data traffic for the subscriber.

Figure 27: DHCP Subscribers in a Fixed Wireless Access Network



References

For detailed information about all aspects of a fixed wireless network, read the 3GPP technical specifications that define everything. [Table 30 on page 394](#) lists the most relevant specifications.

Table 30: 3GPP Technical Specifications for Fixed Wireless Access Networks

Specification Number	Title
3GPP TS 23.002 (Release 15)	Network architecture
3GPP TS 23.401 (Release 15)	General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access
3GPP TS 29.274 (Release 15)	3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C) Stage 3
3GPP TS 29.281 (Release 15)	General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)

3GPP Fixed Wireless Access Terminology

[Table 31 on page 395](#) explains terminology used for 3GPP fixed wireless access networks.

Table 31: Terminology for a 3GPP Fixed Wireless Access Network

Term	Description
3GPP	The 3rd Generation Partnership Project is an international standards organization that develops specifications and protocols for wireless telephony.

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (*Continued*)

Term	Description
APN	<p>The access point name identifies the packet data network (PDN), such as the Internet, that the subscriber wants to access. When a subscriber requests access, the UE passes the requested APN to the eNodeB, which sends it to the MME for authorization. If the subscriber does not request an APN, the MME can authorize a default APN.</p> <p>Each PDN that the user subscribes to has an APN and an associated packet data network gateway (PGW) that the UE uses to access the PDN.</p> <p>The combination of APN and PGW is called a PDN subscription context. One context is the default APN, which always connects to a PDN such as the Internet unless the user activates another APN.</p> <p>The HSS maintains subscriber profiles, The MME uses the profile from the HSS to validate whether the subscriber is actually subscribed to the requested APN.</p> <p>You can also think of the APN as the set of service-level and connection parameters—such as QoS parameters—that is authorized for the UE. A given UE can access many APNs.</p> <p>An APN has two parts:</p> <ul style="list-style-type: none"> • Network Identifier—This defines the external PDN that the user connects to through a PGW. This part of the APN is mandatory. It can be as simple as internet or have a more complicated structure such as example.net. The network identifier can optionally specify a requested subscriber service. <p style="padding-left: 40px;">Operator Identifier—(Optional) This defines the provider whose PDN the user connects to through a PGW. This part of the APN is often omitted. If present, it consists of the provider's Mobile Network Code (MNC) and Mobile Country Code (MCC).</p> <p>An APN that includes both a Network Identifier and an Operator Identifier corresponds to a DNS name for the PGW.</p> <p>The APN has the following format:</p> <p style="padding-left: 40px;"><i>network-id.mncmnc-number.mccmcc-number.gprs</i></p> <p>An APN can be a simple string or more complex, as shown in these examples:</p> <ul style="list-style-type: none"> • fixed-wireless • web.example.net • internet.mnc99.mcc999.gprs

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (*Continued*)

Term	Description
Bearer	<p>A bearer is the tunnel that connects the UE to a PDN through the PGW. A <i>default bearer</i> is established to a default PGW whenever the UE is activated. Activation means here that the UE is on and has performed authentication.</p> <p>A UE device has a default bearer for each PGW to which it connects. For example, if user equipment connects to the Internet through one PGW and a corporate intranet through another PGW, two default bearers will be active.</p> <p>Default bearers are best-effort. The UE can establish <i>dedicated bearers</i> to other PDNs that can have different QoS requirements, such as a guaranteed bit rate (GBR).</p> <p>Bearers encapsulate user data with GTP-U. The GTP-U information is in turn sent over a UDP connection and inside IP packets.</p>
eNodeB	<p>The hardware (typically in a radio tower) that connects directly to the UE over the air and to the wireless network core. Also called evolved Node B or E-UTRAN Node B.</p> <p>The eNodeB has the following functions:</p> <ul style="list-style-type: none"> • Terminates the radio connection from the UE. • Locates the MME that authenticates the UE (SIM card) with information from the subscriber profile maintained on the HSS. • Maintains the S1-U data plane interface with the SAEGW on the BNG. An S1-U interface can support multiple eNodeBs.
GPRS	<p>The general packet radio service is the data standard that defines the specifications that enable wireless networks to carry IP packets to external networks.</p>
GR	<p>The home gateway router that provides the interface between the subscriber's network and the UE. Also called a residential gateway router.</p>
GTP	<p>The GPRS tunneling protocol governs the creation and use of GTP tunnels that carry traffic between two GPRS support nodes (GSNs), such as an MME and an SGW.</p> <p>Each GTP tunnel is identified by a TEID. The receiving end of a tunnel assigns locally the TEID that the transmitting side uses. The tunnel endpoints on the nodes exchange messages to communicate the TEID values to each other.</p>

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (Continued)

Term	Description
GTP-C	The GPRS tunneling protocol, control plane. GTP-C tunnels carry packet data units and signaling messages in the control plane (S11 interface) between tunnel endpoints on the MME and the SAEGW on the BNG.
GTP-U	The GPRS tunneling protocol, user plane. GTP-U tunnels carry packet data units and signaling messages in the user (data) plane (S1-U interface) between tunnel endpoints on the eNodeB and the SAEGW on the BNG.
HSS	<p>The Home Subscriber Server maintains a database of subscriber and service information. This information supports call (connection) control and session management. The HSS has the following functions:</p> <ul style="list-style-type: none"> • Provides authentication information from the subscriber profile to the MME. The MME uses that information to authenticate the UE for the wireless access network connection. • Identifies the APN that represents and defines the connection for the UE.
IMSI	The International Mobile Subscriber Identity number that identifies a 3GPP subscriber. The IMSI consists of a mobile country code, a mobile network code, and a mobile station identification number.
MEI	The Mobile Equipment Identity number that uniquely identifies the subscriber device.

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (Continued)

Term	Description
MME	<p>The mobility management entity is the control node for the wireless access network, communicating with eNodeB, HSS, and SAEGW. Some of its functions include the following:</p> <ul style="list-style-type: none"> • Maintains the S6a interface with the HSS. • Manages and stores contexts for the UE. • Authenticates the UE with the HSS by using various types of UE identification, such as IMSI, MEI, or MSISDN. • Maintains the S11 control plane interface with the SAEGW on the BNG. An S11 interface can support multiple MMEs. • Selects the SAEGW on the BNG for the subscriber session. • Sends messages to the SAEGW for traffic control. • Participates in the bearer activation/deactivation process. • Manages the UE idle state (so the device is reachable from other devices and services), and performs idle-mode paging of UE.
MSISDN	<p>The Mobile Subscriber ISDN Number (telephone number) that is assigned to the mobile subscriber.</p>
PGW	<p>The packet data network gateway provides the UE with connectivity to external networks such as the Internet. Traffic to and from the UE is processed by the PGW. The BNG functions as an SAEGW, which includes the functions of both PGW and SGW.</p> <p>The PGW performs the following functions:</p> <ul style="list-style-type: none"> • Applies the APN characteristics to the UE session. • Allocates IP addresses to the UE during setup of the default bearer. • Filters packets to and from the subscriber. • Enforces policy. • Collects charging information for processing.

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (Continued)

Term	Description
S11	<p>The GTPv2-based control plane interface that connects the MME and the SAEGW on the BNG. GTP-C tunnels carry control messages.</p> <p>An S11 interface can support multiple MMEs.</p>
S1-MME	<p>The GTPv2-based control plane interface that connects eNodeB and the MME.</p>
S1-U	<p>The GTPv1-based user plane interface that connects eNodeB and the SAEGW on the BNG. S1-U is also called the data plane interface. GTP-U tunnels on the interface carry user payloads.</p> <p>An S1-U interface can support multiple eNodeBs.</p>
S6a	<p>Interface that connects the MME and the HSS, which use this interface to exchange subscriber, service, and UE information.</p>
SAEGW	<p>The System Architecture Evolution gateway that includes the functions of both the SGW and the PGW. It enables the BNG to act as both SGW and PGW.</p>
SGW	<p>The serving gateway routes and forwards subscriber data packets. The BNG functions as an SAEGW, which includes the functions of both PGW and SGW.</p> <p>The SGW performs the following functions:</p> <ul style="list-style-type: none"> • Terminates S1-U interfaces with eNodeBs and S11 interfaces with MMEs. • Subscriber management for UE DHCP subscribers.

Table 31: Terminology for a 3GPP Fixed Wireless Access Network (Continued)

Term	Description
TEID	<p>A tunnel endpoint identifier that uniquely identifies a GTP tunnel endpoint in the scope of a path. A fully qualified TEID consists of an IP address concatenated with a locally allocated identifier. Four TEIDs are defined, together they uniquely identify a default bearer session:</p> <ul style="list-style-type: none"> • L-TEID-C consists of the IP address of the S11 interface on the BNG and the BNG's allocated identifier. • L-TEID-U consists of the IP address of the S1-U interface on the BNG and the BNG's allocated identifier • R-TEID-C consists of the IP address of the S11 interface on the MME and the MME's allocated identifier. • R-TEID-U consists of the IP address of the S1-U interface on the eNodeB and the eNodeB's allocated identifier
UE	<p>The user equipment that connects to the wireless network's eNodeB and to the subscriber's network. UE corresponds to what is called CPE in other contexts.</p> <p>In some cases, the UE consists of a SIM card and a residential gateway router (RG) that can host the SIM. In other cases the SIM might be in a separate device that connects to the RG. In both cases, the SIM provides the wireless radio connectivity to eNodeB in the fixed wireless access network.</p>

Benefits of Fixed Wireless Access

- Reduce last-mile installation and maintenance costs by using radio backhaul towers connected to hard-wired network instead of providing fiber to the building.
- Ability to increase service offerings to underserved end users.

How to Configure Fixed Wireless Access

The fixed wireless access configuration enables the SAEGW on the BNG. At a minimum you must configure an APN, the control plane and the data plane.

The following procedure assumes that you have configured separately any of the following that apply for your APN:

- Access profile
- Dynamic profile
- Anchor point interface
- Address pool

To configure fixed wireless access on the BNG:

1. Specify the name of the pseudowire physical interface that anchors all incoming GTP-U (S1-U interface) tunnels on the BNG.

NOTE: The BNG supports only a single anchor point.

```
[edit services fixed-wireless-access]
user@host# set anchor-point anchor-point-name
```

2. Define an access point name for the user equipment by specifying the connection and service parameters that the subscriber's device can use to connect to the PGW to access a PDN.

```
[edit services fixed-wireless-access]
user@host# edit apn access-point-name
```

- a. (Optional) Associate an authentication or accounting profile with the APN that specifies authentication or accounting parameters.

```
[edit services fixed-wireless-access apn access-point-name]
user@host# set aaa-profile aaa-profile-name
```

- b. (Optional) Specify one or more authentication parameters that the BNG sends to the external AAA server for subscribers using this APN. You configure details about the external server in the access profile that you associate with the APN. Some networks might not use this authentication, because the HSS has already authenticated the UE and determined subscriber access. Other networks might not use this because they use Diameter for online charging.

```
[edit services fixed-wireless-access apn access-point-name]
user@host# set authentication password password
user@host# set authentication username-include delimiter delimiter
user@host# set authentication username-include domain-name domain-name
```

```

user@host# set authentication username-include imsi
user@host# set authentication username-include mei
user@host# set authentication username-include msisdn
user@host# set authentication username-include user-prefix user-prefix

```

- c. (Optional) Specify the type of data sent to the APN.

NOTE: Only the ipv4 type is supported.

```

[edit services fixed-wireless-access apn access-point-name]
user@host# set apn-data-type type

```

- d. (Optional) Specify a text description for the APN. You can use this to provide more information about the APN than its name alone can convey. The description for an APN appears in subscriber profiles in the HSS database.

```

[edit services fixed-wireless-access apn access-point-name]
user@host# set description description

```

- e. Associate a dynamic profile with the APN to create the dynamic fixed wireless interface.

NOTE: Services such as CoS and firewall filters are applied as part of the DHCP configuration.

```

[edit services fixed-wireless-access apn access-point-name]
user@host# set dynamic-profile dynamic-profile

```

- f. (Optional) Specify the name of the address pool for IPv4 addresses assigned to the APN.

NOTE: Only IPv4 addresses are supported.

```

[edit services fixed-wireless-access apn access-point-name]
user@host# set ipv4-address-pool pool-name

```

g. (Optional)

```
[edit services fixed-wireless-access apn access-point-name]
user@host# set routing-instance routing-instance-name
```

3. Configure the control plane by specifying the name of the MME and the IPv4 address of the S11 interface on the BNG. The S11 interface is the reference point or connection between the MME and the SAEGW on the BNG for control packets.

```
[edit services fixed-wireless-access]
user@host# set control-plane name s11 v4-address ip-address
```

4. Configure the data plane by specifying the name of the eNodeB and the IPv4 address for the S1-U interface on the BNG. The S1-U interface is the reference point or connection between the eNodeB and the SAEGW on the BNG for subscriber data traffic.

```
[edit services fixed-wireless-access]
user@host# set data-plane name s1-u v4-address ip-address
```

For example, the following configuration specifies the S11 interface address as 192.0.2.30 and the S1-U interface address as 192.0.2.100. The pseudowire anchor is ps0. The APN is named internet-basic; it has both an access profile for RADIUS parameters and a dynamic client profile attached. The authentication password is \$ABC123. The username includes the domain name example.net and the subscriber's IMSI. The data type is the required IPv4. A descriptive string, fwa-basic-subscribers-phenix, is associated with the APN.

The APN uses the default routing instance, because no other routing instance is configured. IP addresses are supplied by RADIUS, because the configuration does not specify a local pool.

```
fixed-wireless-access {
  anchor-point ps0;
  control-plane mme-45 {
    s11 {
      v4-address ip-address;
    }
  }
  data-plane x-s1u-20 {
    s1-u {
      v4-address 192.0.2.100;
    }
  }
}
```



```

}
apn internet-basic {
  aaa-profile fwa-radius-prof;
  apn-data-type ipv4
  authentication {
    password $ABC123;
    username-include {
      domain-name example.net;
      imsi;
    }
  }
  description fwa-basic-subscribers-phoenix;
  dynamic-profile fwa-dyn-profile;
}
}

```

The dynamic profile that you attach to the APN creates the dynamic interface for fixed wireless access. The following configuration is a simple example:

```

dynamic-profiles fwa-dyn-profile {
  interfaces {
    "$junos-interface-ifd-name" {
      unit "$junos-interface-unit" {
        family inet {
          unnumbered-address lo0.0 preferred-source-address 10.0.0.1;
        }
      }
    }
  }
}
}

```

The anchor point interface configuration is also very simple. For example:

```

interfaces {
  ps0 {
    anchor-point {
      rlt-0/1/10;
    }
    flexible-vlan-tagging;
  }
}

```

Verifying and Monitoring Fixed Wireless Access

IN THIS SECTION

- Purpose | 406
- Action | 406

Purpose

Determine status information and statistics for fixed wireless access configurations.

Action

- To display a list of all interfaces on the BNG supporting fixed wireless access subscribers:

```
user@host>show subscribers client-type fixed-wireless-access
```

- To display detailed information about fixed wireless access subscribers, including username and IP address, dynamic profile, local and remote TEIDs, and remote and local IP addresses for the BNG connection to eNodeBs and MMEs:

```
user@host>show subscribers client-type fixed-wireless-access detail
```

- To view statistics for messages exchanged between the BNG, eNodeB, and MME:

```
user@host>show services fixed-wireless-access statistics
```

Release History Table

Release	Description
19.2R1	Starting in Junos OS Release 19.2R1, MX Series routers acting as BNGs can support subscribers in Third-Generation Partnership Project (3GPP) fixed wireless access networks, enabling the integration of fixed wireless subscribers with the wireline subscriber management backend.

RELATED DOCUMENTATION

[Tracing Fixed Wireless Access Events for Troubleshooting | 407](#)

[Configuring Access Profile Options for Interactions with RADIUS Servers](#)

[Dynamic Profiles for Subscriber Management](#)

[Address-Assignment Pools for Subscriber Management](#)

Tracing Fixed Wireless Access Events for Troubleshooting

IN THIS SECTION

- [Configuring the Fixed Wireless Access Trace Log Filename | 408](#)
- [Configuring the Number and Size of Fixed Wireless Access Log Files | 408](#)
- [Configuring Access to the Fixed Wireless Access Log File | 409](#)
- [Configuring a Regular Expression for Fixed Wireless Access Messages to Be Logged | 409](#)
- [Configuring the Fixed Wireless Access Tracing Flags | 410](#)

The Junos OS trace feature tracks fixed wireless access operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `tcpfwdd`. You can specify a different filename, but you cannot change the directory in which trace files are located.
2. When the trace log file `filename` reaches 128 kilobytes (KB), it is compressed and renamed `filename.0.gz`. Subsequent events are logged in a new file called `filename`, until it reaches capacity again. At this point, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the [System Log Explorer](#).)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

The following topics describe how to configure all aspects of tracing fixed wireless access operations:

Configuring the Fixed Wireless Access Trace Log Filename

By default, the name of the file that records trace output for fixed wireless access is `bbe-fwsd`. You can specify a different name with the `file` option.

To configure the filename for fixed wireless access tracing operations:

- Specify the name of the file used for the trace output.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file fwsd_1
```

Configuring the Number and Size of Fixed Wireless Access Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format `.number.gz`. The newest archived file is `.0.gz` and the oldest archived file is `.(maximum number)-1.gz`. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, `filename`, reaches 2 MB, `filename` is compressed and renamed `filename.0.gz`, and a new file called `filename` is created. When the new `filename` reaches 2 MB, `filename.0.gz` is renamed `filename.1.gz` and `filename` is compressed and renamed `filename.0.gz`. This process repeats until there are 20 trace files. Then the oldest file, `filename.19.gz`, is simply overwritten when the next oldest file, `filename.18.gz` is compressed and renamed to `filename.19.gz`.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit system processes tcp-forwarding traceoptions]  
user@host# set file fwsd_1 _logfile_1 files 20 size 2097152
```

Configuring Access to the Fixed Wireless Access Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit system processes tcp-forwarding traceoptions]  
user@host# set file fwsd_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit system processes tcp-forwarding traceoptions]  
user@host# set file fwsd_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for Fixed Wireless Access Messages to Be Logged

By default, the trace operation output includes all messages relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit system processes tcp-forwarding traceoptions]  
user@host# set file fwsd_1 _logfile_1 match regex
```

Configuring the Fixed Wireless Access Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit system processes tcp-forwarding traceoptions]  
user@host# set flag flag-name
```

RELATED DOCUMENTATION

| [Fixed Wireless Access Networks](#) | 390

8

CHAPTER

Configuration Statements and Operational Commands

[Junos CLI Reference Overview](#) | 412

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options 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](#)
- [Operational Commands](#)