Junos® OS

MLPPP Feature Guide for Subscriber Management

Release

14.1

Published: 2014-04-25

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS MLPPP Feature Guide for Subscriber Management
14.1

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 http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- MX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xsl;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xsl; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		    route default {
		      nexthop *address*;
		      retain;
		    }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/

- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

MLPPP for LNS and PPPoE Subscribers

MLPPP Overview

Multilink Point-to-Point Protocol (MLPPP) aggregates multiple PPP physical links into a single virtual connection, or logical bundle. More specifically, MLPPP bundles multiple link-layer channels into a single network-layer channel. Peers negotiate MLPPP during the initial phase of Link Control Protocol (LCP) option negotiation. Each router indicates that it is multilink capable by sending the multilink option as part of its initial LCP configuration request.

An MLPPP bundle can consist of multiple physical links of the same type—such as multiple asynchronous lines—or can consist of physical links of different types—such as leased synchronous lines and dial-up asynchronous lines.

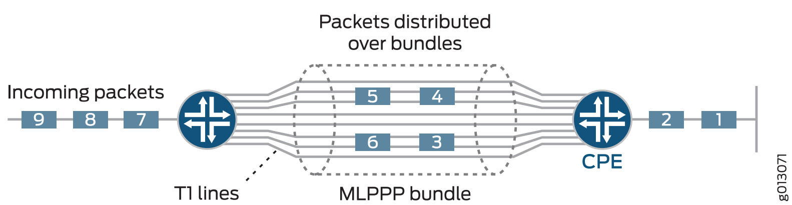
Packets received with an MLPPP header are subject to fragmentation, reassembly, and sequencing. Packets received without the MLPPP header cannot be sequenced and can be delivered only on a first-come, first-served basis.

This section contains the following topics:

Traditional MLPPP Application

MLPPP is used to bundle multiple low speed links to create a higher bandwidth pipe such that the combined bandwidth is available to traffics from all links, and to support link fragmentation and interleaving (LFI) support on the bundle to reduce the transmission delay of high priority packets. LFI interleaves voice packets with fragmented data packets to ensure timely delivery of voice packets. Figure 1 on page 4 shows how incoming packets are distributed and aggregated into an MLPPP bundle.

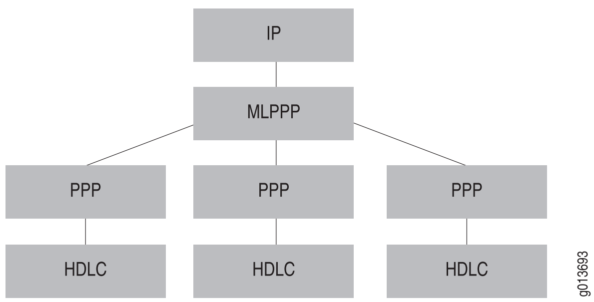
Figure 1: MLPPP Aggregation of Traffic Into Single Bundle



Because MLPPP aggregates multiple link-layer channels onto a single network-layer IP interface, protocol layering within the router is different than for non-multilink PPP.

Figure 2 on page 4 illustrates interface stacking with MLPPP.

Figure 2: Structure of MLPPP



MLPPP LCP Negotiation Option

Multilink PPP adds the multilink maximum received reconstructed unit (MRRU) option for LCP negotiation. The MRRU option has two functions:

- It informs the other end of the link the maximum reassembled size of the PPP packet payload that the router can receive.
- It informs the other end that the router supports MLPPP.

When you enable multilink on your router, the router includes the MRRU option in LCP negotiation with the default value set to 1500 bytes (user-configurable option) for PPP. If the remote system rejects this option, the local system determines that the remote system does not support multilink PPP and it terminates the link without negotiation.

NOTE: **The router does not bring up a link if the MRU value received from a peer device differs from the MRRU value received from the peer.**

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series on page 7
- Understanding MLPPP Link Fragmentation and Interleaving on page 13

MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview

Multilink PPP (MLPPP) support is provided to LNS (L2TP network server) and PPPoE (Point-to-Point Protocol over Ethernet) terminated and tunneled subscribers running on MX Series with access-facing MPC2s.

For customers with both MLPPP and single link PPP clients, the MX Series needs to determine client capability during link control protocol (LCP) negotiation and support either multilink or single link access modules accordingly (mixed mode support).

This section contains the following topics:

Single Member Link MLPPP Bundle Support

MLPPP running on the MX Series provides link fragmentation and interleaving (LFI) support for a single-link bundle. Each bundle contains a single member link only; configuration of multiple member links belonging to the same bundle are rejected. However, LFI enables the single subscriber session to send small, high priority packets interleaving with large packets without introducing unacceptable transmission delay for high priority small packets. LFI interleaves voice packets with fragmented data packets to ensure timely delivery of voice packets and to guarantee voice quality.

Customers with lower bandwidth subscribers benefit from the MLPPP LFI support. With the traditional non-MLPPP application, the CPE (customer premises equipment) device performs the fragmentation prior to the PPP encapsulation and then relies on the application at the far end to perform the reassembly. With the MLPPP solution, the burden to reassemble the packets on the customer servers and the far-end application is removed, and control is given to the service provider for fragmentation and reassembly.

NOTE: A maximum of 8000 MLPPP bundles is supported.

Member Link and Bundle Configuration

An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For MLPPP subscribers, you can configure the member link and bundle statically, or dynamically using dynamic profiles.

- Static MLPPP Subscribers—You must configure both member link and bundle IFLs manually before the member link IFL can start LCP (link control protocol) negotiation either for an LNS session or for a PPPoE session.
- Dynamic MLPPP Subscribers—You configure dynamic member IFLs using dynamic profiles. The member link dynamic profile includes the **family mlppp** statement

containing the bundle dynamic profile and the service interface (**si**), or a pool of service interfaces. This information is then used to create the dynamic bundle IFL.

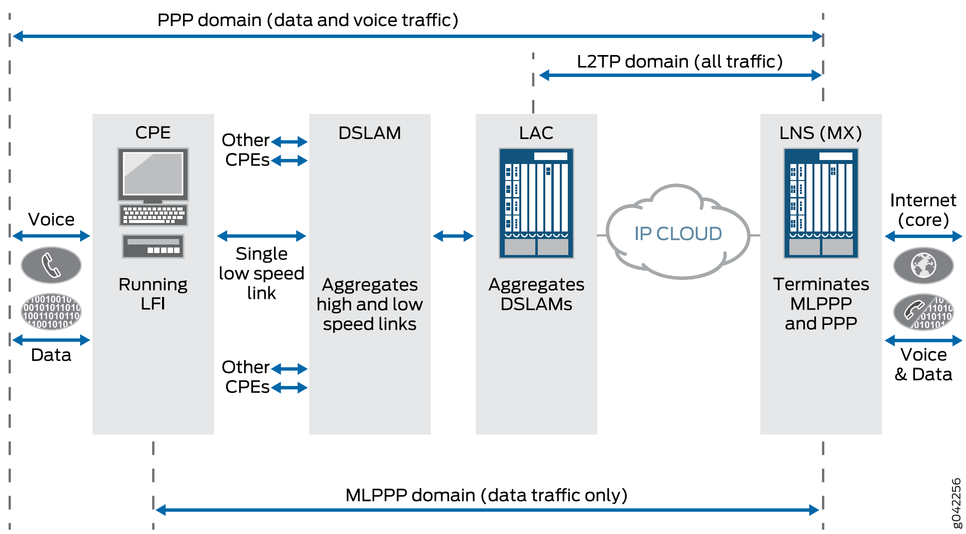
Each bundle accepts only one member link. If more than one member link attempts to join the same bundle, the system fails the new member session.

Dual-stack is supported for the bundle.

LNS Subscribers and MX Series

Figure 3 on page 6 shows a network diagram with the MX Series functioning as the LNS. Both PPP and MLPPP bundles are terminated at the LNS.

Figure 3: MLPPP Bundles Terminated at MX Series as the LNS Network



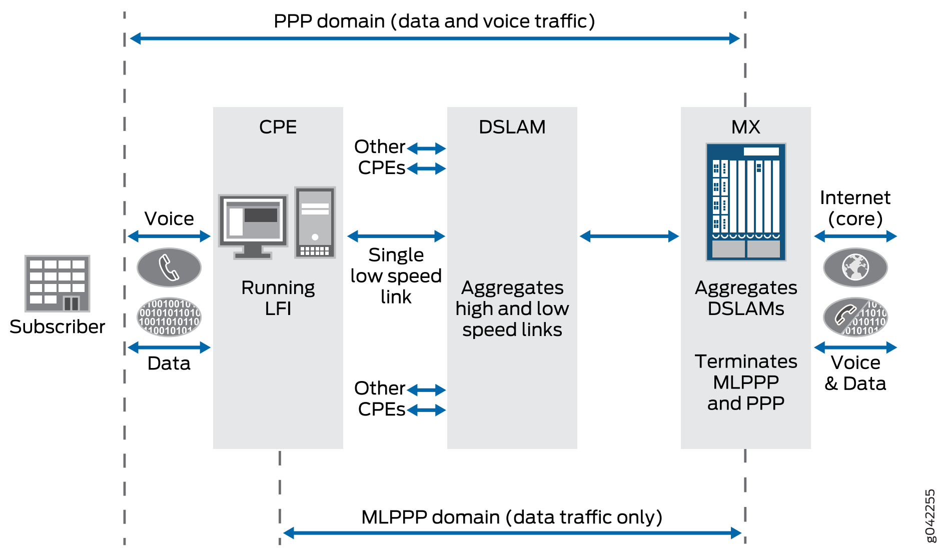
The following three domains are shown passing traffic through the LNS network:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only
- L2TP domain—Contains all types of traffic

PPPoE Subscribers and MX Series

Figure 4 on page 7 shows a network diagram with the MX Series terminating PPPoE sessions that include both the PPP and MLPPP bundles.

Figure 4: PPPoE Sessions Terminated at MX Series



The following two domains are shown passing traffic through the network:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only

Related Documentation

- MLPPP Overview on page 3
- Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series on page 7
- Mixed Mode Support for MLPPP and PPP Subscribers Overview on page 9
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11

Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series

The following features are supported for multilink PPP (MLPPP) for L2TP network server (LNS) and Point-to-Point Protocol over Ethernet (PPPoE, terminated and tunneled) subscribers on the MX Series:

- Supports MLPPP for static and dynamic LNS subscribers and PPPoE subscribers.
- Supports each MLPPP bundle containing a single member link.
- Anchors the bundle logical interface (IFL) on the inline services **si** interface.
- Runs the bundle IFL on an MX Series that enables shaping and queuing at the bundle to minimize fragment reordering.
- Supports configurable service device pools for load-balancing bundle IFLs.

- Supports the co-existence for member link IFL and the bundle IFL on different lookup engines.
- Supports fragmentation maps for both static and dynamic **si** interfaces, and supports multiple forwarding classes pointing to a single queue for **si** interface attachments.
- Provides fragmentation of low-priority packets towards the subscriber, and reassembly of low-priority packets towards the core, and availability of per-bundle fragmentation and reassembly statistics.
- Supports bundle **family inet** and **family inet6**, including DHCPv6 prefix delegation over MLPPP bundle for both LNS and PPPoE MLPPP subscribers.
- Supports lawful intercept over MLPPP bundles.
- Provides mixed mode (PPP and MLPPP) support for subscribers.
- Maintains existing LNS and PPPoE subscriber management functionalities.
- Supports graceful Routing Engine switchover (GRES).

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Mixed Mode Support for MLPPP and PPP Subscribers Overview on page 9
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11

CHAPTER 2

Mixed Mode Support

Mixed Mode Support for MLPPP and PPP Subscribers Overview

Existing customer edge subscriber services separate MLPPP and PPP support for subscribers. However, if a subscriber interface is configured for MLPPP and the customer premises equipment (CPE) does not support MLPPP, then the subscriber login fails.

In an environment where MLPPP and PPP subscribers are mixed and you cannot easily manage the subscriber types by classifying them into separate groups using dynamic profiles, the MX Series needs the capability to renegotiate Link Control Protocol (LCP) in PPP if the CPE rejects LCP negotiation in MLPPP. This capability is known as *mixed mode support*.

Mixed mode uses common configuration and flexibility to support PPP and MLPPP. If you configure a subscriber interface using the **family mlppp** and **family inet/inet6** statements for PPP-only CPE, mixed mode support enables additional LCP negotiation exchanges to successfully negotiate LCP in PPP. Mixed mode supports static and dynamic PPPoE (terminated and tunneled) and LNS (L2TP network server) subscribers.

This section contains the following topics:

PPPoE Terminated and Tunneled Subscribers

If you do not configure the **family mlppp** statement for a subscriber interface, the MX Series negotiates LCP in PPP as it currently does, and any LCP request that contains MLPPP options is rejected.

However for PPPoE subscribers, if you configure the **family mlppp** statement for a subscriber interface, the MX Series negotiates LCP in MLPPP with the CPE. If the CPE rejects MLPPP, then the MX Series renegotiates LCP in PPP with the CPE.

Mixed mode operation for a LAC (tunneled PPPoE) subscriber is the same as for a terminated PPPoE subscriber. The authentication phase has no effect on LAC mixed mode operation because LCP negotiation must be completed prior to authentication.

LNS Subscribers

For LNS subscribers, the MX Series negotiates LCP as follows:

- If proxy data from the LAC indicates that MLPPP was negotiated, and the proxy data is acceptable, and the **lcp-renegotiation** statement is not configured, then the proxy is accepted and the subscriber is MLPPP.
- If proxy data from the LAC indicates that PPP was negotiated, or if there was no proxy data from LAC, or if the **lcp-renegotiation** statement is configured for the LAC, then the MX Series starts LCP negotiation in MLPPP with the CPE.

  If the CPE rejects MLPPP, then the MX Series renegotiates LCP in PPP with the CPE.

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series on page 7
- Configuring L2TP Client Access to Support MLPPP for Static Subscribers on page 39
- Example: Configuring Dynamic LNS MLPPP Subscribers on page 69

CHAPTER 3

MLPPP Bundles and Inline Service Interfaces

MLPPP Bundles and Inline Service Logical Interfaces Overview

Each MLPPP bundle for LNS or PPPoE (terminated and tunneled) subscribers is represented by an inline service (**si**) logical interface (IFL).

This topic contains the following sections:

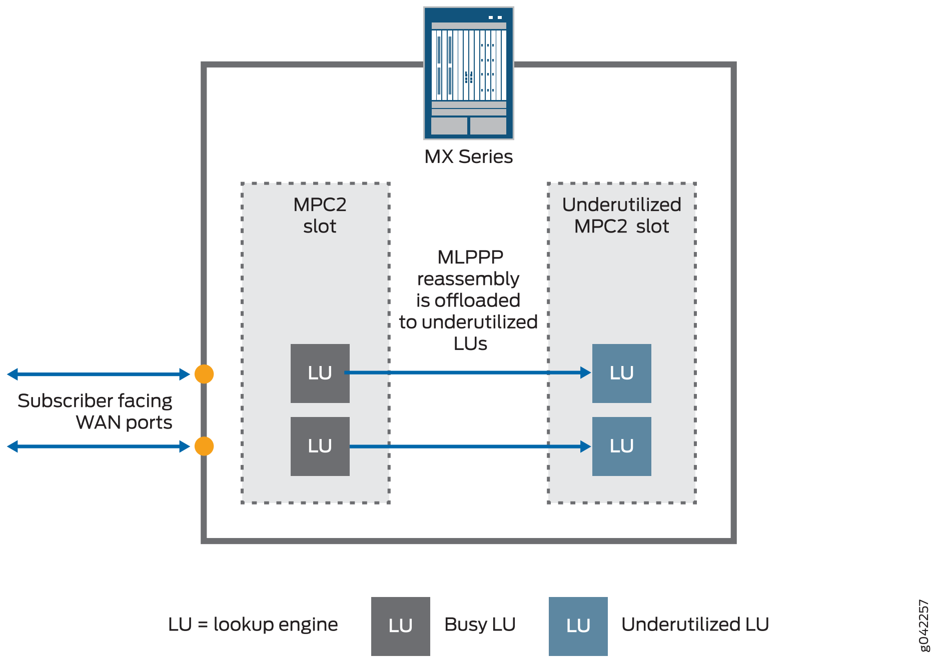
- Distribution of Reassembly Processing on page 11
- Aggregation Point for True Multilink PPP on page 12
- LAC Subscriber Bundle on page 12

Distribution of Reassembly Processing

L2TP network server (LNS) can sustain a throughput of approximately 67 percent of line rate for 64-byte packets. Additionally, MLPPP reassembly must be performed on a subset of these L2TP sessions. By introducing an **si** interface for the bundle, some of the MLPPP reassembly processing can be offloaded to another lookup engine different from the one that is performing the LNS processing.

For example, Figure 5 on page 12 shows a typical MX Series containing two access-facing MPC2 slots, with each slot containing two lookup engines. One or two of the lookup engines are underutilized within the MPC2 slots. The underutilized lookup engines are available to host **si** interfaces to offload MLPPP reassembly processing.

Figure 5: Distribution of MLPPP Reassembly Processing



NOTE: To minimize fragment reordering, the MLPPP si interface must be on an MPC2 where shaping and queuing is performed at the bundle.

Aggregation Point for True Multilink PPP

You can map each link of a multilink bundle to a different lookup engine for LNS processing. Using an **si** interface for the bundle guarantees that all fragments belonging to the same bundle arrive at a single lookup engine for reassembly.

LAC Subscriber Bundle

After a subscriber is tunneled, the bundle is no longer involved in both the control plane and the forwarding path, and both MLPPP bundle IFL and session ID are noted in the graphical user interface.

Related Documentation

- Enabling Inline Service Interfaces for PPPoE and LNS Subscribers on page 34
- Configuring Inline Service Interface for PPPoE and LNS Subscribers on page 35
- Understanding MLPPP Link Fragmentation and Interleaving on page 13

CHAPTER 4

MLPPP Link Fragmentation and Interleaving

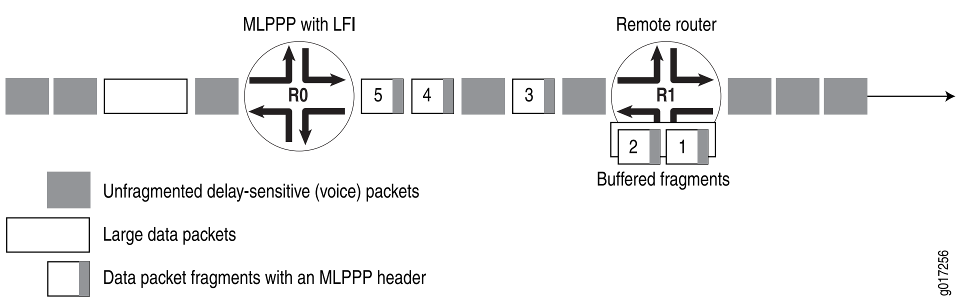
Understanding MLPPP Link Fragmentation and Interleaving

Priority scheduling on a multilink (MLPPP) bundle determines the order in which an output interface transmits traffic from an output queue. The queues are serviced in a weighted round-robin fashion. But when a queue containing large packets starts using the MLPPP bundle, small and delay-sensitive packets must wait their turn for transmission. Because of this delay, some slow links can become useless for delay-sensitive traffic.

Link fragmentation and interleaving (LFI) solves this problem by reducing delay and jitter on links by fragmenting large packets and interleaving delay-sensitive packets with the resulting smaller packets for simultaneous transmission across multiple links of a MLPPP bundle.

Figure 6 on page 13 shows how LFI processes packets.

Figure 6: LFI Packet Processing



Device R0 and Device R1 have LFI enabled. When Device R0 receives large and small packets, such as data and voice packets, it divides them into two categories:

- All voice packets and any other packets configured to be treated as voice packets are categorized as LFI packets and transmitted without fragmentation or an MLPPP header.
- The remaining non-LFI (data) packets are fragmented or unfragmented based on the configured fragmentation threshold. Packets larger than the fragmentation threshold are fragmented. An MLPPP header (containing a multilink sequence number) is added to all non-LFI packets, fragmented and unfragmented.

Fragmentation is performed according to the fragmentation threshold that you configure. For example, if you configure a fragmentation threshold of 128 bytes, all packets greater than 128 bytes are fragmented. When Device R1 receives the packets, it sends the unfragmented voice packets immediately but buffers the packet fragments until it receives the last fragment for a packet. In this example, when Device R1 receives fragment 5, it reassembles the fragments and transmits the whole packet.

The unfragmented data packets are treated as a single fragment. Device R1 transmits the unfragmented data packets as it receives them and does not buffer them.

Related Documentation

- Understanding MLPPP and Fragmentation-Maps on page 14
- Understanding Fragmented Packet Queuing on page 17
- Understanding Sequenced Packet Fragment Drops on page 21

Understanding MLPPP and Fragmentation-Maps

You enable link fragmentation and interleaving (LFI) on inline service (**si**) interface bundles by configuring **fragmentation-maps**. For multilink PPP (MLPPP) bundle support, you must configure **fragmentation-maps** in **class-of-services** and reference them in either the bundle dynamic-profile or bundle logical interface (IFL) configuration.

BEST PRACTICE: For MX Series and class-of-service (CoS) implementation, you can configure a fragmentation map to have two forwarding classes pointing to the same queue. However, if you assign multiple forwarding classes to a single queue, you must also reference all of those forwarding classes in a fragmentation map to enable the expected behavior.

If you reference only one of the forwarding classes assigned to a queue, then the other forwarding classes in that queue can clog that queue with large packets. For previous existing fragmentation-map implementations, this condition did not occur because the other forwarding classes inherited this fragmentation behavior assigned to that queue.

If you assign multiple forwarding classes to a queue, create a fragmentation map that addresses each of those forwarding classes. This results in fragmentation-map behavior that more closely reflects the expected behavior based on the fragmentation CLI, while the existing fragmentation-map behavior remains unchanged.

This section contains the following topics:

Fragmentation-Map Settings

By setting **fragmentation-maps** under **class-of-service**, you can configure the fragmentation properties on a particular forwarding class, as shown in the following sample output:

```
class-of-service {
 fragmentation-maps {
  map-name {
   forwarding-class class-name {
    fragment-threshold bytes;
    no-fragmentation;
   }
  }
 }
}
```

You can configure the following settings for **fragmentation-maps**:

- (Optional) **fragment-threshold**—Sets a per-forwarding class fragmentation threshold in bytes. **fragment-threshold** sets the maximum size of each multilink fragment. An extra MLPPP header is prepended to these multilink fragments. This same header is also prepended to packets of these forwarding classes that are smaller than the fragmentation threshold.
    - For MLPPP bundle interface configuration, you can set the **fragment-threshold** for all forwarding classes. Any fragmentation threshold defined by a **fragmentation-map** and applied to that interface takes precedence for the forwarding classes referenced by that **fragmentation-map**.
    - For **si** bundle IFL configuration, the **fragment-threshold** applies to all forwarding classes. The **fragment-threshold** setting in **fragmentation-maps** for a particular forwarding class, if configured, overrides the threshold configured in **si** bundle IFL for that class. If no **fragment-threshold** is configured anywhere, packets are still fragmented if the threshold exceeds the smallest MTU or MRRU of all links in the bundle.
- (Required) **no-fragmentation**—Sets traffic on a particular forwarding class to be interleaved rather than fragmented. The **no-fragmentation** setting is required to define high priority traffic and indicates that an extra fragmentation header is not prepended to the packets of this forwarding class

NOTE: For a given forwarding class, you can include either the **fragment-threshold** setting or the **no-fragmentation** setting; they are mutually exclusive.

Understanding Fragmentation-Map Bindings

Using MLPPP in this manner generates two subscriber interfaces for each subscriber:

- The inline services (**si**) bundle interface IFL.
- The PPP member link IFL.

The data plane traffic destined for the subscriber exits through the (**si**) bundle interface IFL, and passes through the PPP member link IFL. Queuing is provided for both of these IFLs, which then requires the ability to define class of service.

When you are creating the two subscriber interfaces, the MX Series authenticates only a single user, and the RADIUS server only provides a single set of class-of-service (CoS) attributes. These CoS RADIUS attributes are then applied to both the (**si**) bundle interface IFL and the PPP member link IFL.

NOTE: For this scenario to succeed, you must have already configured the dynamic profiles for these IFLs to accept CoS RADIUS attributes enabling both the (**si**) bundle interface IFL and the PPP member link IFL to have the same CoS attributes.

To apply different CoS to the (**si**) bundle interface IFL and the PPP member link IFL, you can set CoS RADIUS attributes to specify the Transmission Control Protocol (TCP) name to which the attribute is intended. The dynamic profile associated with the (**si**) bundle interface IFL contains the CoS TCP for that IFL, and the dynamic profile associated with the PPP member link IFL contains the CoS TCP for that IFL.

The RADIUS attributes each include a target TCP. When configured, two sets of CoS RADIUS attributes are retrieved with the member link authentication; one set with the (**si**) bundle interface IFL TCP specified, and the other set with the PPP member link IFL TCP specified.

Related Documentation

- *fragmentation-maps*
- Understanding MLPPP Link Fragmentation and Interleaving on page 13
- Understanding Fragmented Packet Queuing on page 17
- Understanding Sequenced Packet Fragment Drops on page 21

Understanding Fragmented Packet Queuing

Fragmented Multilink PPP (MLPPP) packets have a multilink header containing a multilink sequence number. The sequence numbers on these fragments must be preserved so that the remote device receiving these fragments can correctly reassemble them into a complete packet. To accommodate this requirement, Junos OS queues all packets on member links of a multilink bundle with a MLPPP header into a single queue (q0) by default.

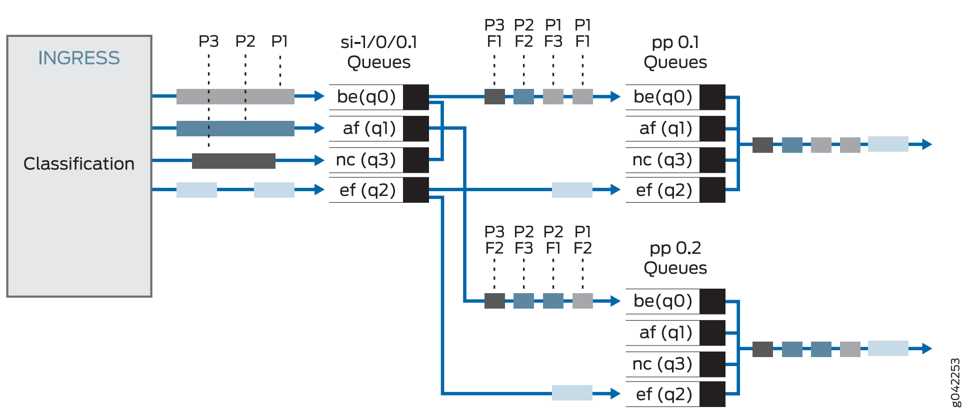
- Traffic flows of a forwarding class that has MLPPP fragmentation configured are distributed from the inline services **si** bundle interface queues to the member link queues (queue 0) following a round-robin method.
- Traffic flows of a forwarding class without MLPPP fragmentation are distributed from the **si** bundle interface queues to the member link queues based on a hashing algorithm computed from the destination address, source address, and IP protocol of the packet.

  If the IP payload contains TCP or UDP traffic, the hashing algorithm also includes the source and destination ports. As a result, all traffic belonging to one traffic flow is queued to one member link.

Figure 7 on page 18 shows how traffic is queued on an MLPPP multilink bundle and its member links. Packet flows in the figure use the notation P*x*,F*x*; for example, P1,F1 represents Packet 1, Fragment 1.

- There are four queues.
- Forwarding classes be, af, and nc are mapped to queues q0, q1, and q3, respectively, on the multilink bundle. These are fragmented.
- Forwarding class ef contains voice traffic, and is mapped to q2 and is not fragmented.
- Interface **si-1/0/0.1** is the bundle, and **pp0.1** and **pp0.2** are the member links for that bundle.

Figure 7: Queuing on Member Links



Queuing on member links proceeds as follows:

1. The packet fragments of forwarding classes be, af, and nc on the multilink bundle are mapped to q0 on Member Links 1 and 2. These packets are distributed from the **si** queues to the member links using a round-robin method.
2. The packets of forwarding class ef (voice) from the multilink bundle are mapped to q2 on the member links. This forwarding class is not fragmented. The packets are distributed from the **si** queues to the member links based on a hashing algorithm.
3. The network control packets from the multilink bundle are mapped to q0 on the member links. The bundle network control traffic is queued with the data flows on the member link. However, q3 on the member links transmits network control packets that exchange protocol information related to member links, such as packets exchanging hello messages on member links.

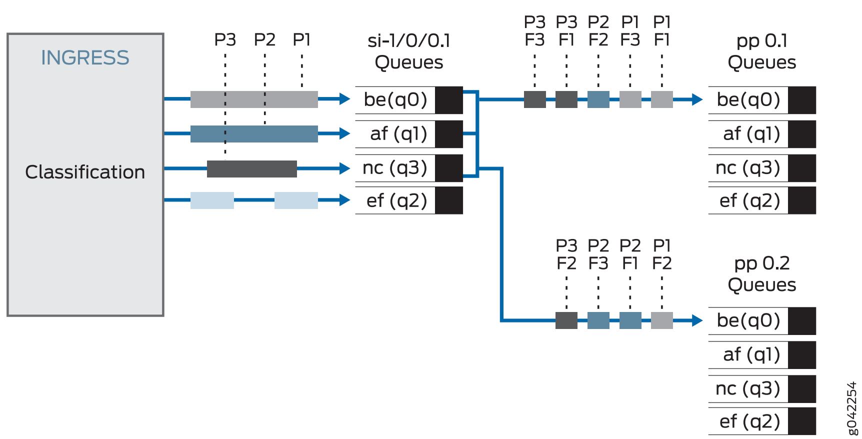
This section contains the following topics:

Queuing of Fragmented Packets to Member Links

On a multilink bundle, packet fragments from all forwarding classes with fragmentation enabled are transmitted to q0 on member links. On the q0 queues of member links, packets are queued using a round-robin method to enable per-fragment load balancing.

Figure 8 on page 19 shows how fragmented packet queuing is performed on the member links. Packet flows in the figure use the notation P*x*,F*x*; for example, P1,F1 represents Packet 1, Fragment 1.

Figure 8: Queuing of Fragmented Packets on Member Links



Packet fragments from the multilink bundle are queued to member links one by one using a round-robin method:

- Packet P1,F1 from q0 on the multilink bundle is queued to q0 on Member Link 1.
- Packet P1,F2 from q0 on the multilink bundle is queued to q0 on Member Link 2.

- Packet P1,F3 from q0 on the multilink bundle is queued to q0 on Member Link 1.
- Packet P2,F1 from q1 on the multilink bundle is queued to q0 on Member Link 2, and so on.

NOTE: **Packets that are part of the fragmented forwarding class, but are not fragmented, follow the same procedure.**

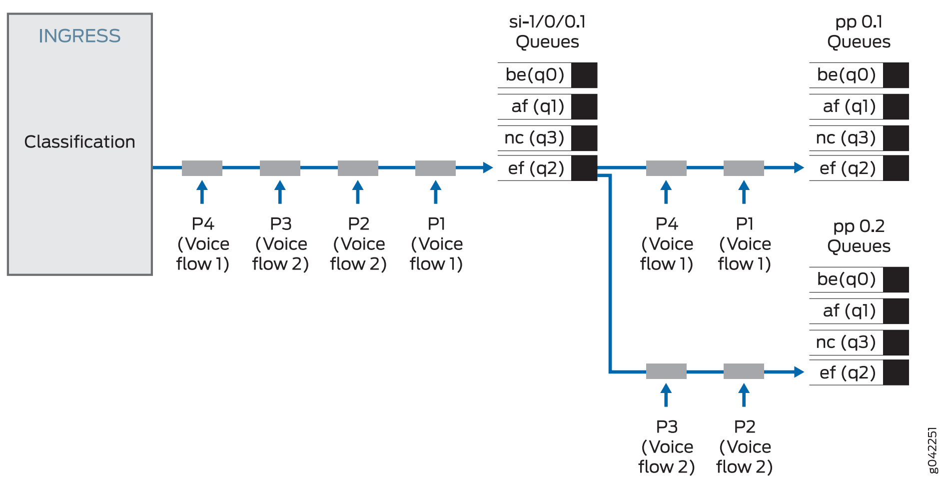
After exiting the **si** interface, Microcode adds a header of approximately 40 bytes to the MLPPP packets. When configuring the class-of-service shaping, you may need to adjust bytes to account for this.

Queuing of LFI Packets to Member Links

On a multilink bundle, all non-MLPPP encapsulated traffic [link fragmenting and interleaving (LFI) traffic] from the multilink bundle are queued to the queue as defined by the forwarding class of that packet.

Figure 9 on page 20 shows how LFI packet queuing is performed on the member links.

Figure 9: Queuing of LFI Packets on Member Links



The packets are distributed from the **si** interface to the member links based on a hashing algorithm computed from the source address, destination address, and IP protocol of the packet.

If the IP payload contains TCP or UDP traffic, the hashing algorithm also includes the source and destination ports. As a result, all traffic belonging to one traffic flow is queued to one member link.

Related Documentation

- Understanding MLPPP Link Fragmentation and Interleaving on page 13
- Understanding MLPPP and Fragmentation-Maps on page 14
- Understanding Sequenced Packet Fragment Drops on page 21

Understanding Sequenced Packet Fragment Drops

Multilink PPP (MLPPP) link fragmentation and interleaving (LFI) provides buffering at the receiver side of a link to reassemble MLPPP fragmented packets. Dropping of the packet fragments is a concern because the packet fragments' remainder consumes valuable bandwidth and buffer space, only to have it eventually being dropped.

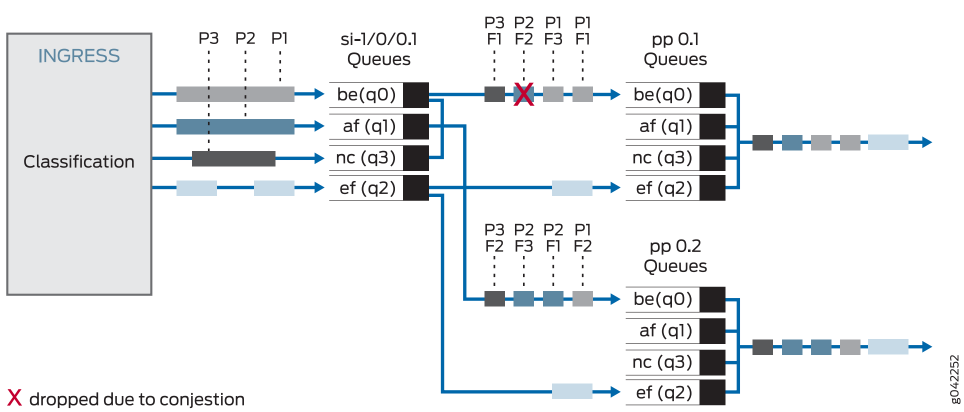
The MX Series provides two stages of queuing for packets exiting an MLPPP bundle:

- The first stage of queuing is performed at the inline services **si** interface.
- The second stage is performed by adding member link scheduler queues.

During the first stage of queuing at the **si** interface, when exiting from these queues, LFI packets are fragmented and assigned a sequence number. These fragmented packets are then distributed to the member links where they are queued for the second time.

Congestion at the member link queues can result in MLPPP packet fragments being dropped, as shown in Figure 10 on page 21. Packet flows in the figure use the notation P*x*,F*x*; for example, P1,F1 represents Packet 1, Fragment 1.

Figure 10: Dropped Sequenced Packet Fragment



Data packet and fragment P2,F2 is dropped due to congestion at the **pp0.1** queues. This occurs after the sequence numbers have been assigned for packet P2.

In a Broadband Remote Access Server (B-RAS) implementation, the bundle member links share the physical interface with other bundle member links, as well as with PPP subscriber interfaces, causing the physical interface to be oversubscribed and most likely creating congestion.

During the second stage of queuing, member link scheduler queues are added to provide a degree of protection against the port traffic congestion causing fragmented MLPPP packets to be dropped. See Figure 11 on page 22 and Figure 12 on page 22 for member link scheduler hierarchies.

NOTE: All MLPPP packets are sent to queue 0 (be).

Figure 11: si Bundle Interface Scheduler Hierarchy

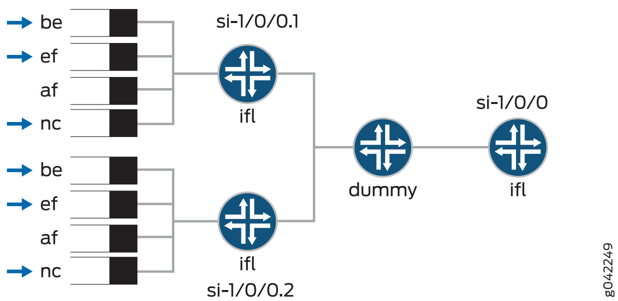
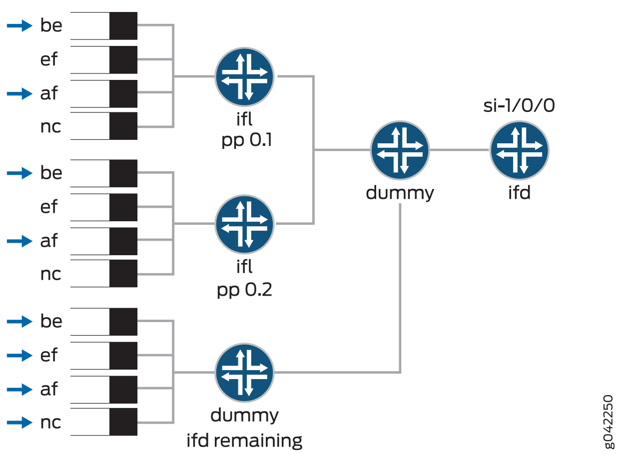


Figure 12: MLPPP Member Link Scheduler Hierarchy



Related Documentation

- Understanding MLPPP Link Fragmentation and Interleaving on page 13
- Understanding MLPPP and Fragmentation-Maps on page 14
- Understanding Fragmented Packet Queuing on page 17

CHAPTER 5

MLPPP Subscriber Accounting Statistics

MLPPP Subscriber Accounting Statistics Overview

For broadband subscriber management edge router Point-to-Point Protocol (PPP) subscribers, the accounting statistics contain two groups:

- The aggregate (IPv4 and IPv6) statistics group consists of statistics reported through these RADIUS attributes: **Acct-Input-Octets**, **Acct-Output-Octets**, **Acct-Input-Packets**, and **Acct-Output-Packets**.
- The IPv6 portion of the aggregate statistics group reported through the Juniper Networks **ERX-VSA**s **151** through **156**.

Broadband subscriber management edge router PPP logical interfaces (IFLs) support accurate accounting statistics by excluding PPP control traffic, and incrementing packet and octets at the point where the packet is about to leave the router. The packet is not dropped by CoS, filters, or policers.

For MLPPP subscribers, accounting is performed for each member link (currently limited to one) and not the bundle. The bundle IFL supports accurate accounting statistics only, and the member link supports transit statistics only. As a result, the following restrictions apply for member link final aggregate statistics:

- Only aggregate statistics are available with no IPv6 specific statistics; for example, **ERX-VSA 151** to **156** are all zeros.
- Packets sent and received over the member link include fragments and non-fragmented packets.
- Octets sent and received are bytes in the fragments and non-fragmented packets.
- Aggregate statistics include packets that can be dropped in the router, such as CoS, filters, and policers.
- Aggregate statistics include PPP control packets (LCP, PAP, CHAP, and NCP) and keepalive packets.

The following topics describe the statistics collection process in the lookup engine for member links and its bundle.

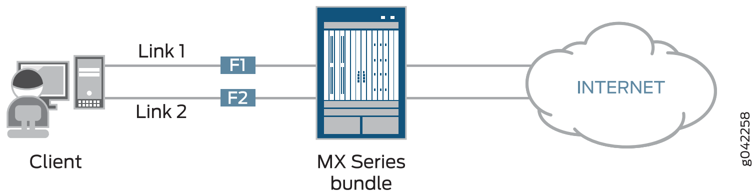
- Member Link and Bundle Statistics Collection on page 24
- RADIUS Final Statistics Output Example on page 26

Member Link and Bundle Statistics Collection

MLPPP with MPC2 currently supports only one member link per bundle. However, support for accounting statistics must consider a true multilink scenario where multiple member links exist per bundle. From the lookup engine, only the bundle has the ability to maintain Layer 3 statistics. For an individual member link, only protocol-agnostic fragments (plus non-fragmented packets) are counted.

Figure 13 on page 25 shows an MLPPP client with two active member links and the statistics maintained by the lookup engine. For MLPPP with MPC2, each member link and bundle can reside on different lookup engines from where the accounting statistics are maintained.

Figure 13: MLPPP Client with Two Active Member Links



Client-to-Internet Traffic Statistics

When the client sends IP packets towards the Internet, they may be fragmented. For example, packet P1 is fragmented into F1 and F2, and the fragments belonging to a single packet can be sent on different links (Figure 13 on page 25).

- F1 is sent on Link 1
- F2 is sent on Link 2

When Link 1 on the MX Series receives fragment F1, it is identified as an MLPPP encapsulated fragment. Because IPv4 or IPv6 families are indicated on the first fragment, all of the incoming fragments are counted using a protocol-agnostic method before the fragment is forwarded to the bundle for reassembly.

- The protocol-agnostic incoming packet count is incremented by 1.
- The protocol-agnostic incoming byte count is incremented by the size of the fragment.

Similarly on Link 2, fragment F2 is also counted using a protocol-agnostic method, and then forwarded to the bundle for reassembly.

Fragment F1 arrives at the bundle and is stored along with its MLPPP header containing the sequence number with the **begin flag** set to 0, and the **end flag** set to 1.

Fragment F2 arrives at the bundle and is stored along with its MLPPP header containing the sequence number with the **begin flag** set to 1, and the **end flag set** to 0.

The pattern of monotonically increasing sequence numbers, **begin flag** set to 1 and **end flag** set to 1, causes fragments F1 and F2 to be reassembled into a single packet.

After the packet has been reassembled, the packet's Layer 3 type (either IPv4 or IPv6) is determined at the bundle. Then, the packets and bytes are counted according to its Layer 3 type at the bundle based on accurate accounting statistics:

- *bundleA_ipv4_packets_from_client* += 1
- *bundleA_ipv4_bytes_from_client* += *packet_size*

Or

- *bundleA_ipv6_packets_from_client* += 1

- *bundleA_ipv6_bytes_from_client* += *packet_size*

Internet-to-Client Traffic Statistics

In the reverse direction, Layer 3 packets come from the Internet to the bundle.

The packets and bytes are counted according to its Layer 3 type at the bundle:

- *bundleA_ipv4_packets_to_client* += 1
- *bundleA_ipv4_bytes_to_client* += *packet_size*

Or

- *bundleA_ipv6_packets_to_client* += 1
- *bundleA_ipv6_bytes_to_client* += *packet_size*

If the packets are fragmented, the fragments belonging to the same packet can be sent out different links. Because no IPv4 or IPv6 families are indicated on the links, all of the outgoing fragments are counted using a protocol-agnostic method.

- The protocol-agnostic outgoing packet count is incremented by 1.
- The protocol-agnostic outgoing byte count is incremented by the size of the fragment.

RADIUS Final Statistics Output Example

The following output example shows RADIUS final statistics:

```
User-Name = "mlUser1@isp1.com"
    Acct-Status-Type = Stop
    Acct-Session-Id = "786"
    Acct-Multi-Session-Id = "787"
    Acct-Input-Octets = 1068151928
    Acct-Output-Octets = 4268692096
    Acct-Session-Time = 61965
    Acct-Input-Packets = 406636696
    Acct-Output-Packets = 357477811
    Acct-Terminate-Cause = Lost-Carrier
    Service-Type = Framed-User
    Framed-Protocol = PPP
    Framed-IPv6-Pool = "v6-pool-21"
    Acct-Authentic = RADIUS
    Acct-Delay-Time = 0
    ERX-Dhcp-Mac-Addr = "0090.1a41.ec2d"
    Event-Timestamp = "Oct 19 2012 10:31:03 IST"
    Framed-IP-Address = 21.0.0.3
    Framed-IP-Netmask = 255.0.0.0
    ERX-Input-Gigapkts = 0
    Acct-Input-Gigawords = 6
    NAS-Identifier = "kalka"
    NAS-Port = 306184213
    NAS-Port-Id = "ge-1/1/9.21:21"
    NAS-Port-Type = Ethernet
    ERX-Output-Gigapkts = 0
```



```
Acct-Output-Gigawords = 4
ERX-Attr-151 = 0x00000000
ERX-Attr-152 = 0x00000000
ERX-Attr-153 = 0x00000000
ERX-Attr-154 = 0x00000000
ERX-Attr-155 = 0x00000000
ERX-Attr-156 = 0x00000000
NAS-IP-Address = 111.1.1.2
Acct-Unique-Session-Id = "03eeef735aef3520"
Timestamp = 1350604541
Request-Authenticator = Verified
```

Related Documentation

- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11
- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series on page 7

CHAPTER 6

Dynamic PPP Subscriber Services

Dynamic PPP Subscriber Services for Static MLPPP Interfaces Overview

Dynamic subscriber services are supported for MLPPP bundle interfaces, with certain interface and hardware restrictions. See "Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces" on page 30. Multiclass MLPPP enables the relative prioritization of up to eight classes of traffic over an MLPPP bundle, but only on link services intelligent queuing (IQ) (LSQ) interfaces.

RADIUS previously supported only authentication for MLPPP. Address management, service deactivation, and dynamic selection of subscriber properties based on RADIUS user ID are now also supported.

RADIUS can dynamically allocate IPv4 addresses for MLPPP connections. When the first subscriber logs in, an address is allocated. The same address is allocated to all links in a bundle. Any other address provided for any of the links is ignored. The IP address is released for re-allocation when the last member link in a bundle logs out. Similar to the address allocation, the services configured for the first subscriber to log in are configured for all subsequent subscribers in the bundle.

The Acct-Multi-Session-Id [50] attribute enables RADIUS to link multiple related sessions into a single log file. RADIUS uses the session database (SDB) bundle session ID for the value of Acct-Multi-Session-Id. This bundle ID enables RADIUS to initiate a disconnect for an entire bundle. By tracking the member link sessions, RADIUS is also able to disconnect the individual member links in a bundle.

The Acct-Link-Count [51] attribute records the number of links present in a multilink session at the time the accounting record is generated.

Related Documentation

- Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30
- Configuring PPP Subscriber Services for MLPPP Bundles on page 105

Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces

PPP subscriber services are supported for MLPPP bundle interfaces. These services require the following hardware:

- M120 router or M320 router
- Channelized DS3/E3 Enhanced IP PIC (PB-4CHDS3-E3-IQE-BNC) to support MLPPP subscriber access
- An Adaptive Services PIC or Multiservices PIC to support subscriber services on LSQ MLPPP bundle interfaces

Subscriber services are not supported for single-link PPP interfaces with this hardware.

Related Documentation

- Dynamic PPP Subscriber Services for Static MLPPP Interfaces Overview on page 29

PART 2

Configuration

CHAPTER 7

Inline Service Interfaces Configuration for PPPoE and LNS Subscribers

Enabling Inline Service Interfaces for PPPoE and LNS Subscribers

The inline service (**si**) interface is a virtual physical interface that resides on lookup engines. The **si** interface, referred to as an *anchor* interface, makes it possible to support multilink PPP (MLPPP) bundles without a special services PIC. The **si** interface is supported on MLPPP on the MX Series.

Four inline service interfaces are configurable per MPC-occupied chassis slot. The following MPC2 slots are supported:

- The MPC2-3D contains two lookup engines, each with two **si** interfaces.
- The PC1-3D contains only one lookup engine and it hosts all four **si** interfaces.

You can configure the following inline service interfaces as anchor interfaces for MLPPP bundles: **si-slot/0/0**, **si-slot/1/0**, **si-slot/2/0**, and **si-slot/3/0**.

- For MLPPP over PPPoE subscribers, **family mlppp** is supported in **pp0** member link IFL, and the bundle is an **si** IFL.
- For MLPPP over LNS subscribers, **family mlppp** is supported in **si-** member link IFL, and the bundle is an **si** IFL.

You enable inline services for PICs 0 to 3 individually by setting the **inline-services** statement at the **[edit chassis]** hierarchy level for the FPCs.

The following example shows how to enable inline services for PIC 0 on MPC slot 1, and PIC 1 on MPC on slot 5, and set 10g as the bandwidth for tunnel traffic. As a result, both **si-1/0/0** and **si-5/0/0** are created for the specified PICs as well.

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 specify the amount of bandwidth reserved on each lookup engine for tunnel traffic using inline services.

   [edit chassis fpc ***slot-number*** **pic** ***number*****]**
   user@host# **set inline-services** ***bandwidth***

The following shows sample output:

```
chassis {
    fpc 1 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
    }
    fpc 5 {
        pic 1 {
```

```
                inline-services {
                    bandwidth 10g;
                }
            }
        }
    }
```

Related Documentation

- Configuring Inline Service Interface for PPPoE and LNS Subscribers on page 35
- Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers on page 36
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11

Configuring Inline Service Interface for PPPoE and LNS Subscribers

The inline service (**si**) interface is a virtual physical interface that resides on lookup engines. The **si** interface, referred to as an *anchor* interface, makes it possible to support multilink PPP (MLPPP) bundles without a special services PIC. The **si** interface is supported on MLPPP on the MX Series. Four inline service interfaces are configurable per MPC-occupied chassis slot.

For existing Layer 2 and Layer 3 services, the **si** interface **unit 0** is currently used to store the unilist next-hop information. However, you must reserve and configure **si** interface **unit 0** and set **family inet** for both PPPoE and LNS subscribers because the **si** interface implements the bundle functionality. Setting **family inet6** is ignored by the system.

The following example shows how to configure inline services for PIC 0 on MPC slot 1, and PIC 1 on MPC on slot 5, and set **unit 0 family inet** for both.

To configure inline service interfaces:

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. Reserve and configure the IPv4 family (**inet**) on the reserved **unit 0** logical interface for PPPoE and LNS subscribers and bundle functionality.

   [edit interfaces si-*slot/pic/port*]
   user@host# **set unit 0 family inet**

The following shows sample output:

```
interfaces {
    si-1/0/0 {
        hierarchical-scheduler maximum-hierarchy-levels 2;
        encapsulation generic-services;
        unit 0 {
            family inet;
        }
    }
    si-5/1/0 {
        hierarchical-scheduler maximum-hierarchy-levels 2;
        encapsulation generic-services;
        unit 0 {
            family inet;
        }
    }
}
```

Related Documentation

- Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers on page 36
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11
- Enabling Inline Service Interfaces for PPPoE and LNS Subscribers on page 34

Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers

With dynamic L2TP network server (LNS) configuration, you can replace the **services-interfaces** with a **service-device-pool** in the tunnel-group for load balancing LNS subscribers. Optionally, you can use the **service-device-pool** statement for load balancing to dynamically select the inline services (**si**) interface for both bundle (PPPoE or LNS subscribers), and LNS member link, respectively.

NOTE: The service-device-pool configuration enables interface overlap, which can result in over usage of the overlapped interfaces.

Before you begin, enable the inline service interfaces for all FPC slots and PICs. See "Enabling Inline Service Interfaces for PPPoE and LNS Subscribers" on page 34.

The following example shows how to configure two service device pools (pool1 and pool2) for inline services for load balancing bundle and LNS member link.

To configure two service device pools:

1. Create the tunnel group.

   [edit services l2tp]
   user@host# **set tunnel-group** ***name***

2. Define the service device pools to assign **si** interfaces for load balancing.

   [edit services l2tp]
   user@host# **set service-device-pool** ***pool-name***

The following shows sample output when all referenced FPC slots and PICs had been enabled for inline services:

```
services {
    service-device-pools {
        pool pool1 {
            interface si-1/0/0;
            interface si-1/1/0;
            interface si-3/0/0;
        }
        pool pool2 {
            interface si-1/1/0;
            interface si-2/1/0;
            interface si-5/1/0;
        }
    }
}
```

Related Documentation

- Configuring Inline Service Interface for PPPoE and LNS Subscribers on page 35
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11
- Example: Configuring Dynamic LNS MLPPP Subscribers on page 69

CHAPTER 8

L2TP Access Client Configuration for MLPPP Over LNS Support

Configuring L2TP Client Access to Support MLPPP for Static Subscribers

To enable MLPPP over LT2P network server (LNS) support for MX Series, you must indicate whether MLPPP is supported for static subscribers from a particular L2TP client (LAC) by configuring the **multilink** statement currently supported in **access profile**. 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). You can configure multiple access profiles and multiple clients within each profile.

With mixed mode support, the **multilink** statement enables MLPPP but does not set it. However, if you do not configure the **multilink** statement, MLPPP is not supported for static LAC subscribers.

The following two examples show L2TP access profile configurations for an MLPPP-capable static L2TP client and non-multilink (single link) static L2TP client.

To configure an L2TP access profile for MLPPP-capable static L2TP clients:

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***

3. 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***

4. Configure the LNS to renegotiate the link control protocol (LCP) with the PPP client.

   [edit access profile *access-profile-name* client *client-name*] user@host# **set l2tp lcp-renegotiation**

5. 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***

6. 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***

7. (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**

8. Specify that the L2TP client is MLPPP-capable for static subscribers.

   [edit access profile *access-profile-name* client *client-name*]
   user@host# **set l2tp multilink**

MLPPP is first negotiated with static subscribers coming from the LAC peer group profile, **ce-lac-1-gp**, but then switches to PPP if the subscriber rejects MLPPP. The following shows sample output for MLPPP-capable static L2TP client:

```
access profile {
        ce-l2tp-profile1 {
        client ce-lac-1 {
            user-group-profile ce-lac-1-gp;
            l2tp {
                interface-id not-used;
                lcp-renegotiation;
                maximum-sessions-per-tunnel 2000;
                shared-secret "$9$2wgUHQF/9pB";
                aaa-access-profile ce-aaa-profile;
                multilink;
            }
        }
    }
}
```

To configure an L2TP access profile for non-MLPPP, or single link static L2TP clients, repeat Step 1 through Step 7 for configuring an L2TP access profile for multilink-capable static L2TP clients. Do not **set l2tp multilink**.

Only PPP is negotiated with static subscribers from the LAC peer group profile, **ce-lac-2-gp**, and an LCP configuration request from the customer premises equipment (CPE) with maximum received reconstructed unit (MRRU) option is rejected. The following shows sample output for single link static L2TP client:

```
access profile {
        ce-l2tp-profile1 {
        client ce-lac-2 {
            user-group-profile ce-lac-1-gp;
            l2tp {
                interface-id not-used;
                maximum-sessions-per-tunnel 1000;
                shared-secret "$9$2aBcXyz/21P";
                aaa-access-profile ce-aaa-profile;
```



```
                                                  ## multilink not entered,
static subscriber is single link only
            }
        }
    }
}
```

Related Documentation

- Mixed Mode Support for MLPPP and PPP Subscribers Overview on page 9
- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Example: Configuring Static LNS MLPPP Subscribers on page 43

Configuring L2TP Client Access to Support MLPPP for Dynamic Subscribers

To enable support for MLPPP over LT2P network server (LNS) on MX Series, you configure the **family mlppp** statement in the **dynamic profile** name, which indicates that MLPPP is supported for dynamic subscribers from a particular L2TP client (LAC).

NOTE: **The multilink statement used to enable MLPPP for static LNS subscribers is ignored for dynamic LNS subscribers if it is configured.**

You can configure a dynamic profile name for the LAC using **access profile** from the **l2tp** statement. If you specify a dynamic profile name in the L2TP client access profile, it overrides the **dynamic-profile** ***name*** specified in the **tunnel-group** used to create the dynamic subscriber interface. If you do not configure a dynamic profile name in the L2TP client access profile, then the **dynamic-profile** ***name*** specified in the **tunnel-group** is used.

The following example shows an L2TP access profile configuration with a dynamic profile name for dynamic LNS subscribers.

To configure an L2TP access profile configuration with a dynamic profile name for dynamic LNS subscribers:

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***

3. 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***

4. 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***

5. 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***

6. (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**

7. Specify the dynamic profile name for the dynamic LNS subscriber.

   [edit access profile *access-profile-name* client *client-name*]
   user@host# **set l2tp dynamic-profile *name***

If the **family mlppp** statement is configured in **dynamic-profile**, MLPPP is negotiated first; otherwise, only PPP is negotiated. The following shows sample output for an L2TP access profile configuration with a dynamic profile name for dynamic LNS subscribers:

```
access profile {
        ce-l2tp-profile2 {
        client ce-lac-3 {
            user-group-profile ce-lac-1-gp;
            l2tp {
                interface-id not-used;
                maximum-sessions-per-tunnel 2000;
                shared-secret "$9$2wgUHQF/9pB";
                aaa-access-profile ce-aaa-profile;
                dynamic-profile ml-lns-member-prof;
            }
        }
    }
}
```

Related Documentation

- *Configuring a Dynamic Profile for Dynamic LNS Sessions*
- Example: Configuring Dynamic LNS MLPPP Subscribers on page 69
- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5

CHAPTER 9

Static MLPPP Subscribers Configuration for MX Series Examples

Example: Configuring Static LNS MLPPP Subscribers

This example shows how to configure static L2TP network server (LNS) multilink (MLPPP) subscribers.

Requirements

This example uses the following hardware and software components:

- MX Series with MPC2s installed
- Junos OS Release 13.3 or later

Before you configure static L2TP network server (LNS) multilink (MLPPP) subscribers, be sure you have:

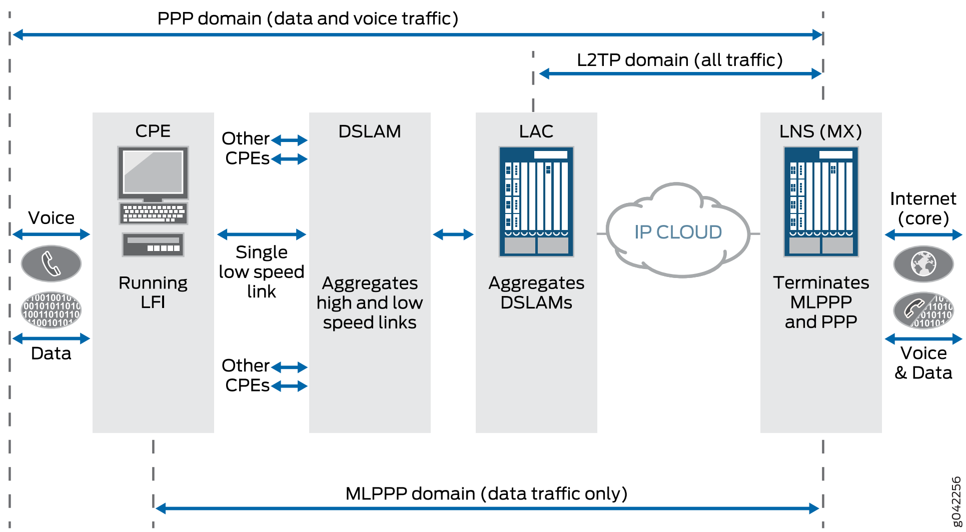
- Enabled the inline service (**si**) interface for LNS subscribers. See "Enabling Inline Service Interfaces for PPPoE and LNS Subscribers" on page 34.
- Configured the inline service (**si**) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 35.

Overview

An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For static MLPPP subscribers, you configure the member link and bundle statically. For static LNS MLPPP subscribers, you configure both member link and bundle IFLs manually. After you configure the subscriber's interface using the **family mlppp** setting, before the member link IFL can start LCP (link control protocol) negotiation for an LNS, you must also fully configure the member link's bundle IFL. Figure 14 on page 44 shows how the different types of traffic traverse through a network where the MX Series device is acting as the LNS to terminate MLPPP bundles.

Topology

Figure 14: MLPPP Bundles Terminated at MX Series as the LNS Network



The following three domains are shown passing traffic through the LNS network:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only
- L2TP domain—Contains all types of traffic

Configuration

To configure static L2TP network server (LNS) multilink (MLPPP) subscribers, perform these tasks:

- Configuring a Tunnel Group with Inline Service Interface and L2TP Access Profile Attributes on page 45
- Configuring a Static LNS Member Link IFL on page 47
- Configuring a Static Inline Services MLPPP Bundle IFL on page 48
- Results on page 49

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.

```
[edit]
```

```
set access profile ce-l2tp-profile1 client ce-lac-1 user-group-profile ce-lac-1-gp
set access profile ce-l2tp-profile1 client ce-lac-1 l2tp lcp-renegotiation
set access profile ce-l2tp-profile1 client ce-lac-1 l2tp maximum-sessions-per-tunnel 2000
set access profile ce-l2tp-profile1 client ce-lac-1 l2tp shared-secret “password”
set access profile ce-l2tp-profile1 client ce-lac-1 l2tp multilink
set services l2tp tunnel-group lns1 l2tp-access-profile ce-l2tp-profile1
set services l2tp tunnel-group lns1 aaa-access-profile ce-authenticator
set services l2tp tunnel-group lns1 local-gateway address 100.1.1.2
set services l2tp tunnel-group lns1 service-interface si-1/0/0

[edit]
set interfaces si-1/0/0.1
set interfaces si-1/0/0.1 dial-options l2tp-interface-id not used dedicated
set interfaces si-1/0/0.1 family mlppp bundle si-5/1/0.100
set interfaces si-1/0/0.1 family inet unnumbered-address lo0.0
set interfaces si-1/0/0.2
set interfaces si-1/0/0.2 dial-options l2tp-interface-id not used dedicated
set interfaces si-1/0/0.2 family mlppp bundle si-5/1/0.101
set interfaces si-1/0/0.2 family inet

[edit]
set interfaces si-5/0/0 unit 100
set interfaces si-5/0/0 unit 100 encapsulation multilink-ppp
set interfaces si-5/0/0 unit 100 mrru 1500
set interfaces si-5/0/0 unit 100 fragment-threshold 640
set interfaces si-5/0/0 unit 100 short-sequence
set interfaces si-5/0/0 unit 100 ppp-options dynamic-profile l2l3-service-prof
```

Configuring a Tunnel Group with Inline Service Interface and L2TP Access Profile Attributes

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure a tunnel group with inline service interface (**si**) and L2TP access profile attributes for static LNS MLPPP subscribers:

1. Create the access profile.

   ```
   [edit access]
   user@host# set profile ce-l2tp-profile1
   ```

2. Configure an L2TP (LAC) access client.

   ```
   [edit access profile ce-l2tp-profile1]
   user@host# set client ce-lac-1
   ```

3. Associate a group profile containing PPP attributes to apply for the PPP sessions being tunneled from this LAC client.

   ```
   [edit access profile ce-l2tp-profile1 client ce-lac1ce-lac1]
   user@host# set user-group-profile ce-lac-1-gp
   ```

4. Configure the following L2TP access profile attributes for this example:

- Link control protocol (LCP) with the PPP client.
- Maximum number of sessions allowed in a tunnel from the client (LAC).
- Tunnel password used to authenticate the client (LAC).
- L2TP client is MLPPP-capable for static subscribers. The **multilink** statement determines whether MLPPP is supported for subscribers coming in from the LAC peer.

[edit access profile ce-l2tp-profile1 client ce-lac1ce-lac1]
user@host# **set l2tp lcp-renegotiation**
user@host# **set l2tp maximum-sessions-per-tunnel 2000**
user@host# **set l2tp shared-secret *password***
user@host# **set l2tp multilink**

NOTE: Do not specify a dynamic profile name in the L2TP access client profile for static LNS MLPPP subscribers.

5. Create the tunnel group.

   [edit services l2tp]
   user@host# **set tunnel-group lns1**

6. Set the tunnel access profile equal to the setting you defined for the access profile.

   [edit services l2tp tunnel-group lns1]
   user@host# **set l2tp-access-profile ce-l2tp-profile1**

7. Set the L2TP AAA access profile.

   NOTE: You can specify the L2TP AAA access profile at either the **[edit access]** or **[edit services]** hierarchy levels, using the LNS access client profile or **tunnel-group** statements, respectively. An L2TP AAA access profile defined using the **[edit access]** hierarchy level overrides the L2TP AAA access profile defined for the **tunnel-group** using the **[edit services]** hierarchy level.

   [edit services l2tp tunnel-group lns1]
   user@host# **set aaa-access-profile ce-authenticator**

8. Set the local gateway address for the L2TP tunnel.

   [edit services l2tp tunnel-group lns1]
   user@host# **set local-gateway address 100.1.1.2**

9. Specify the inline services interface (**si**) for the static LNS MLPPP subscribers.

   [edit services l2tp tunnel-group lns1]
   user@host# **set service-interface si-1/0/0**

10. If you are done configuring the device, commit the configuration.

    [edit]
    user@host# **commit**

Configuring a Static LNS Member Link IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the static LNS member link IFL, you specify the static bundle using the **family mlppp** statement.

You must also configure the **family inet** statement in the subscriber (**si**) interface. The **family inet** setting enables the L2TP long route to be installed and supported for the lookup engine to steer control packets to the Routing Engine; and also enables mixed mode support, if required.

The following example shows that both PPP and MLPPP subscribers can log in successfully using the **si-1/0/0.1** interface, whereas only MLPPP subscribers can log in successfully using the **si-1/0/0.2** interface.

1. Create the **si-1/0/0.1** and **si-1/0/0.2** interfaces.

    [edit interfaces]
    user@host# **set si-1/0/0.1**
    user@host# **set si-1/0/0.2**

2. For the **si-1/0/0.1** interface, set the L2TP dial options to specify that the logical interface can host one session at a time (dedicated).

    [edit interfaces si-1/0/0.1]
    user@host# **set dial-options l2tp-interface-id not used dedicated**

3. Enable MLPPP support and configure the static bundle inline interface (IFL).

    [edit interfaces si-1/0/0.1]
    user@host# **set family mlppp bundle si-5/1/0.100**

4. Enable LNS support and mixed mode support.

    [edit interfaces si-1/0/0.1]
    user@host# **set family inet unnumbered-address lo0.0**

5. For the **si-1/0/0.2** interface, set the L2TP dial options to specify that the logical interface can host one session at a time (dedicated).

    [edit interfaces si-1/0/0.2]
    user@host# **set dial-options l2tp-interface-id not used dedicated**

6. Enable MLPPP support and configure the static bundle inline interface (IFL).

    [edit interfaces si-1/0/0.2]
    user@host# **set family mlppp bundle si-5/1/0.101**

7. Enable LNS long route support.

[edit interfaces si-1/0/0.2]
user@host# **set family inet**

8. If you are done configuring the device, commit the configuration.

   [edit]
   user@host# commit

Configuring a Static Inline Services MLPPP Bundle IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the static inline services (**si**) interface MLPPP bundle IFL, you specify the **encapsulation multilink-ppp** statement within the **si** interface. The **si** interface anchors the bundle interface.

You can also set these optional MLPPP parameters: MRRU, short sequence, and fragment-threshold. The following example shows how to configure the static (**si**) interface MLPPP bundle IFL.

1. Create the static (**si**) interface MLPPP bundle IFL **si-5/0/0** with a unit of 100.

   [edit interfaces]
   user@host# **set si-5/0/0 unit 100**

2. Configure the **encapsulation multilink-ppp** statement to enable MLPPP bundling for the **si-5/0/0.100** interface.

   [edit interfaces si-5/0/0.100]
   user@host# **set encapsulation multilink-ppp**

3. Configure the following MLPPP options for this example:

   - **mrru**—Specifies the maximum received reconstructed unit value ranging from 1500 through 4500 bytes.
   - **fragment-threshold**—Applies to all packets and forwarding classes, ranging from 128 through 16,320 bytes.
   - **short-sequence**—Determines the header format for the MLPPP. Default is **long-sequence**.

   [edit interfaces si-5/0/0.100]
   user@host# **set mrru 1500**
   user@host# **set fragment-threshold 640**
   user@host# **set short-sequence**

4. Enable support for static (**si**) interface IFL dynamic services by configuring the **ppp-options dynamic profile** setting.

   [edit interfaces si-5/0/0.100]
   user@host# **set ppp-options dynamic-profile l2l3-service-prof**

5. If you are done configuring the device, commit the configuration.

   [edit]

user@host# **commit**

Results

From configuration mode, confirm your configuration by entering the **show access**, **show services**, and **show interfaces** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show access profile ce-l2tp-profile1
  access profile {
     ce-l2tp-profile1 {
       client ce-lac-1 {
         user-group-profile ce-lac-1-gp;
         l2tp {
           interface-id not-used;
           lcp-renegotiation;
           maximum-sessions-per-tunnel 2000;
           shared-secret "$9$2wgUHQF/9pB";
           multilink;
         }
       }
     }
  }
```

```
user@host# show services l2tp tunnel-group lns1
  services l2tp {
    tunnel-group lns1 {
      l2tp-access-profile ce-l2tp-profile1;
      aaa-access-profile ce-authenticator;
      local-gateway {
        address 100.1.1.2;
      }
      service-interface si-1/0/0;
    }
  }
```

```
user@host# show interfaces si-1/0/0
  interfaces {
    si-1/0/0 {
      unit 1 {
        dial-options {
          l2tp-interface-id not-used;
          dedicated;
        }
        family mlppp {
           bundle si-5/1/0.100;
        }
        family inet {
          unnumbered-address lo0.0;
        }
      }
      unit 2 {
        dial-options {
          l2tp-interface-id not-used;
          dedicated;
        }
```

```
            family mlppp {
                bundle si-5/1/0.101;
            }
            family inet;
        }
    }
}
```

```
user@host# show interfaces si-5/1/0
interfaces {
    si-5/1/0 {
        unit 100 {
            encapsulation multilink-ppp;
            mrru 1500;
            fragment-threshold 640;
            short-sequence;
            ppp-options {
                dynamic-profile l2l3-service-prof;
            }
        }
    }
}
```

Verification

Confirm that the configuration is working properly.

Verifying the Inline Services Interface Information

Purpose Verify that the inline services (**si**) interface is configured.

Action root@haverhill> **show interfaces si-1/0/0 extensive**

```
Physical interface: si-1/0/0, Enabled, Physical link is Up
  Interface index: 143, SNMP ifIndex: 569, Generation: 146
  Type: Adaptive-Services, Link-level type: Adaptive-Services, MTU: 9192,
  Clocking: Unspecified, Speed: 10000mbps
  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   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                 6068                    0 bps
   Output bytes  :              1072104                  352 bps
   Input  packets:                  126                    0 pps
   Output packets:                12185                    0 pps
   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
```

Meaning The (**si**) interface is enabled with its physical link up and running with Point-to-Point interface flags set. It is shared between LNS subscribers, LNS MLPPP member links, and MX Series MLPPP bundles.

Verifying the Bundle IFL Information

Purpose Verify that the bundle IFL information is correct for MLPPP over LNS subscribers.

Action `root@haverhill>` **show interfaces si-5/1/0.1073756926 extensive**

```
Logical interface si-5/1/0.1073756926 (Index 102) (SNMP ifIndex 607)
(Generation 167)
 Flags: Up Point-To-Point SNMP-Traps 0x84000 Encapsulation: Multilink-PPP
 Last flapped: 2011-04-08 14:13:21 PDT (00:41:48 ago)
 Bandwidth: 10000mbps
 Bundle links information:
   Active bundle links       1
   Removed bundle links      0
   Disabled bundle links     0
 Bundle options:
   MRRU                                 1504
   Remote MRRU                          1504
   Drop timer period                    0
   Inner PPP Protocol field compression disabled
   Sequence number format               long (24 bits)
   Fragmentation threshold              500
   Links needed to sustain bundle       1
   Interleave fragments                 Enabled
   Multilink classes                    0
   Link layer overhead                  4.0 %
 Bundle status:
   Received sequence number        0xffffff
   Transmit sequence number        0xffffff
   Packet drops                    0 (0 bytes)
   Fragment drops                  0 (0 bytes)
   MRRU exceeded                   0
   Fragment timeout                0
   Missing sequence number         0
   Out-of-order sequence number    0
   Out-of-range sequence number    0
   Packet data buffer overflow     0
   Fragment data buffer overflow   0
 Statistics          Frames        fps         Bytes          bps
 Bundle:
   Multilink:
     Input :             3          0            270           0
     Output:             3          0            285           0
   Network:
     Input :             3          0            252           0
     Output:             3          0            276           0
 IPV6 Transit Statistics         Packets       Bytes
   Network:
     Input :                      0              0
     Output:                      0              0
 Link:
   si-1/0/0.1073756925
     Up time: 00:06:37
     Input :           126          0           9596           0
     Output:           126          0           1226           0
 Multilink detail statistics:
 Bundle:
   Fragments:
     Input :             0          0              0           0
     Output:             0          0              0           0
   Non-fragments:
     Input :             0          0              0           0
     Output:             0          0              0           0
   LFI:
```



```
      Input :             0          0              0             0
      Output:             0          0              0             0
 NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
   Protocol inet, MTU: 1500, Generation: 154, Route table: 0
     Flags: Sendbcast-pkt-to-re
     Addresses, Flags: Is-Primary
       Destination: Unspecified, Local: 80.80.80.1, Broadcast: Unspecified,
Generation: 150
```

Meaning Due to the particulars of implementation, the following error counts associated with a bundle always display 0: packet drops (bytes), fragment drops (bytes), fragment timeout, missing sequence number, out-of-order sequence number, out-of-range sequence number, packet data buffer overflow and fragment data buffer overflow, and MRRU exceeded.

Verifying the Member Link IFL Information

Purpose Verify that the member link IFL information is correct for subscribers.

Action `root@haverhill>` **show interfaces si-1/0/0.1073756925 extensive**

```
  Logical interface si-5/1/0.1073756925 (Index 80) (SNMP ifIndex 3286)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: Adaptive-Services
    Last flapped: 2011-04-08 14:13:21 PDT (00:41:48 ago)
    Traffic statistics:
     Input  bytes  :                228
     Output bytes  :                  0
     Input  packets:                  3
     Output packets:                  0
    Local statistics:
     Input  bytes  :                228
     Output bytes  :                  0
     Input  packets:                  3
     Output packets:                  0
    Transit statistics:
     Input  bytes  :                  0                    0 bps
     Output bytes  :                  0                    0 bps
     Input  packets:                  0                    0 pps
     Output packets:                  0                    0 pps
Protocol mlppp, Multilink bundle: si-5/1/0.1073756926
Service interface: si-1/0/0, Dynamic profile: ml-bundle-prof
MTU: 9188, Generation: 15538, Route table: 0
```

Meaning Multilink bundle **si-5/1/0.1073756926** has been configured using the family **mlppp** protocol.

Verifying the Subscriber Information

Purpose Verify that the subscriber information for static MLPPP over LNS is correct.

Action `root@haverhill>` **show subscribers extensive**

```
  Type: L2TP
User Name: client1@rtb.com
IP Address: 80.80.80.10
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: si-1/0/0.1
Interface type: Static
State: Active
Radius Accounting ID: 1
Session ID: 1
Bundle Session ID: 2
Login Time: 2011-04-11 07:55:59 PDT

Type: MLPPP
User Name: client1@rtb.com
IP Address: 80.80.80.10
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: si-5/1/0.100
Interface type: Static
State: Active
Radius Accounting ID: 2
Session ID: 2
Underlying Session ID: 1
Login Time: 2011-04-11 07:55:59 PDT
```

Meaning Subscriber information for interface **si-5/1/0.100** has been configured for MLPPP with interface type of static.

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Configuring L2TP Client Access to Support MLPPP for Static Subscribers on page 39
- Example: Configuring Static PPPoE MLPPP Subscribers on page 54

Example: Configuring Static PPPoE MLPPP Subscribers

This example shows how to configure static Point-to-Point Protocol over Ethernet (PPPoE) MLPPP for terminated and tunneled subscribers.

- Requirements on page 54
- Overview on page 55
- Configuration on page 56
- Verification on page 62

Requirements

This example uses the following hardware and software components:

- MX Series with MPC2s installed

- Junos OS Release 13.3 or later

Before you configure static PPPoE MLPPP for terminated and tunneled subscribers, be sure you have:

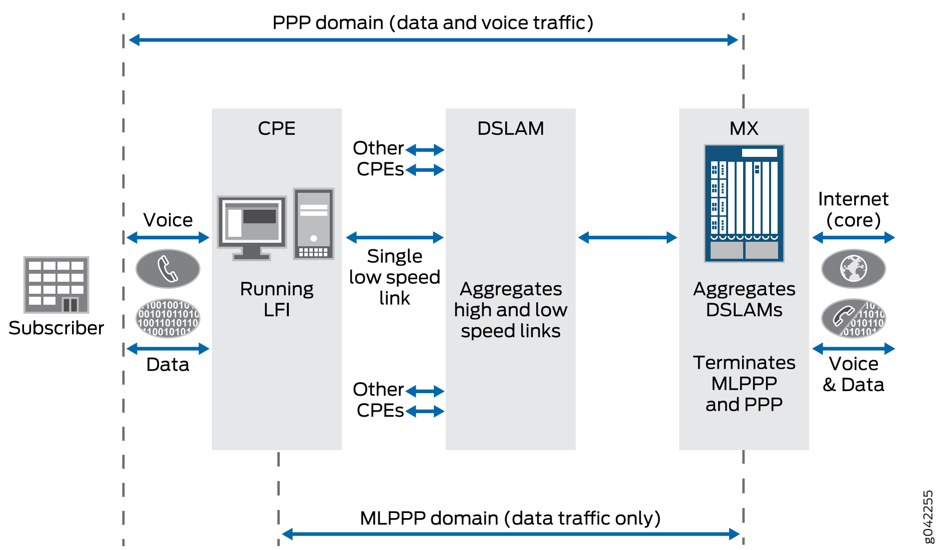
- Enabled the inline service (**si**) interface for LNS subscribers. See "Enabling Inline Service Interfaces for PPPoE and LNS Subscribers" on page 34.
- Configured the inline service (**si**) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 35.

Overview

An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For static MLPPP subscribers, you configure both member link and bundle IFLs manually. After you configure the subscriber's interface using the **family mlppp** statement, before the member link IFL can start LCP (link control protocol) negotiation PPPoE session, you must also fully configure the member link's bundle IFL. Figure 15 on page 55 shows how the different types of traffic traverse through a network where the MX Series terminates PPPoE sessions.

Topology

Figure 15: PPP and MLPPP Traffic Terminated at MX Series



The following two domains are shown terminating traffic at the MX Series:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only

Configuration

To configure static PPPoE MLPPP for terminated and tunneled subscribers, perform these tasks:

- Configuring a Static pp0 Member Link IFL on page 57
- Configuring a Static Inline Services MLPPP Bundle IFL on page 60
- Results on page 61

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.

```
[edit]
set interfaces ge-3/0/0 vlan-tagging
set interfaces ge-3/0/0 unit 1 encapsulation ppp-over-ether vlan-id 1
set interfaces ge-3/0/0 unit 2 encapsulation ppp-over-ether vlan-id 2
set interfaces ge-3/0/0 unit 3 encapsulation ppp-over-ether vlan-id 3
set interfaces pp0
set interfaces pp0 unit 1 keepalives interval 30
set interfaces pp0 unit 1 pppoe-options underlying interface ge-3/0/0.1 server
set interfaces pp0 unit 1 ppp-options pap chap dynamic-profile pp0-l2l3-service prof
set interfaces pp0 unit 1 family mlppp bundle si-1/0/0.1
set interfaces pp0 unit 1 family inet unnumbered-address lo0.0
set interfaces pp0 unit 1 family inet6 address 2040:2004::1.1.1.2/64
set interfaces pp0 unit 2 keepalives interval 30
set interfaces pp0 unit 2 pppoe-options underlying-interface ge-3/0/0.2 server
set interfaces pp0 unit 2 ppp-options pap dynamic-profile pp0-l2l3-service prof
set interfaces pp0 unit 2 family mlppp bundle si-1/0/0.2
set interfaces pp0 unit 3 keepalives interval 30
set interfaces pp0 unit 3 pppoe-options underlying interface ge-3/0/0.3 server
set interfaces pp0 unit 3 ppp-options pap chap dynamic-profile pp0-l2l3-service prof
set interfaces pp0 unit 3 family mlppp bundle si-1/0/0.3
set interfaces pp0 unit 3 family inet
```

```
[edit]
set interfaces si-5/0/0 unit 100
set interfaces si-5/0/0 unit 100 encapsulation multilink-ppp
set interfaces si-5/0/0 unit 100 mrru 1500
set interfaces si-5/0/0 unit 100 fragment-threshold 640
set interfaces si-5/0/0 unit 100 short-sequence
set interfaces si-5/0/0 unit 100 ppp-options dynamic-profile l2l3-service-prof
```


Configuring a Static pp0 Member Link IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the static PPPoE member link, you specify the static bundle using the **family mlppp** statement. PPPoE sessions are supported over the following underlying interfaces: Ethernet interfaces, static and dynamic VLAN, VLAN demultiplexing (demux) over Ethernet interfaces, and VLAN demux over aggregated Ethernet interfaces.

You must also configure the **family inet** statement in the **pp0** interface for tunneled subscribers. The **family inet** statement enables the L2TP long route to be installed and supported for the lookup engine to steer control packets to the Routing Engine.

The following example shows how to configure **pp0** member link IFL over static VLAN to support the following different types of subscribers:

- **si-1/0/0.1**—Both terminated and tunneled PPP and MLPPP subscribers can log in successfully.
- **si-1/0/0.2**—Only terminated MLPPP subscribers can log in successfully.
- **si-1/0/0.3**—Terminated and tunneled MLPPP subscribers can log in successfully.

1. Create the Gigabit Ethernet underlying interface for the PPPoE session, **ge-3/0/0**, and enable VLAN tagging.

   [edit interfaces]
   user@host# **set ge-3/0/0 vlan-tagging**

2. For the **ge-3/0/0** interface, configure PPP over Ethernet encapsulation for three VLANs.

   [edit interfaces ge-3/0/0]
   user@host# **set unit 1 encapsulation ppp-over-ether vlan-id 1**
   user@host# **set unit 2 encapsulation ppp-over-ether vlan-id 2**
   user@host# **set unit 3 encapsulation ppp-over-ether vlan-id 3**

3. Configure the dynamic PPPoE **pp0** subscriber interface to support PPPoE sessions.

   [edit interfaces]
   user@host# **set pp0**

4. Configure the first of three logical interfaces.

   a. Configure the first logical interface for the **pp0** subscriber interface on the MX Series and set an interval of 30 seconds for the keepalive value.

      [edit interfaces pp0]
      user@host# **set unit 1 keepalives interval 30**

   b. Configure the underlying interface **ge-3/0/0.1** and PPPoE server mode for a dynamic PPPoE logical interface in a dynamic profile.

      [edit interfaces pp0 unit 1]
      user@host# **set pppoe-options underlying-interface ge-3/0/0.1 server**

c. Configure PPP-specific interface properties in a dynamic profile: **pap** and **chap**, and set the **dynamic-profile** to the services dynamic profile.

NOTE: The dynamic profile is applied when Link Control Protocol (LCP) is negotiated in PPP.

[edit interfaces pp0 unit 1]
user@host# **set ppp-options pap chap dynamic-profile pp0-l2l3-service prof**

d. Configure the static bundle for the PPPoE member link for MLPPP subscribers using the **family mlppp** statement.

NOTE: The **family mlppp** statement determines whether MLPPP is supported for subscribers coming in from the underlying interface.

[edit interfaces pp0 unit 1]
user@host# **set family mlppp bundle si-1/0/0.1**

e. Configure the **family inet** statement and the unnumbered address for the protocol family required for PPP subscribers for tunneled PPP and for MLPPP subscribers.

[edit interfaces pp0 unit 1]
user@host# **set family inet unnumbered-address lo0.0**

f. (Optional) Enable the **family inet6** statement and address for the mixed mode support for PPP and MLPPP subscribers.

[edit interfaces pp0 unit 1]
user@host# **set family inet6 address 2040:2004::1.1.1.2/64**

5. Configure the second of three logical interfaces.

a. Configure the second logical interface for the **pp0** subscriber interface on the MX Series and set an interval of 30 seconds for the keepalive value.

[edit interfaces pp0]
user@host# **set unit 2 keepalives interval 30**

b. Configure the underlying interface **ge-3/0/0.2** and PPPoE server mode for a dynamic PPPoE logical interface in a dynamic profile.

[edit interfaces pp0 unit 2]
user@host# **set pppoe-options underlying interface ge-3/0/0.2 server**

c. Configure PPP-specific interface properties in a dynamic profile: **pap**, and set the **dynamic-profile** to the services dynamic profile.

NOTE: The dynamic profile is applied when Link Control Protocol (LCP) is negotiated in PPP.

[edit interfaces pp0 unit 2]
user@host# **set ppp-options pap dynamic-profile pp0-l2l3-service prof**

d. Configure the static bundle for the PPPoE member link for MLPPP subscribers using the **family mlppp** statement.

NOTE: The **family mlppp** statement determines whether MLPPP is supported for subscribers coming in from the underlying interface.

[edit interfaces pp0 unit 2]
user@host# **set family mlppp bundle si-1/0/0.2**

6. Configure the last of three logical interfaces.

a. Configure the third logical interface for the **pp0** subscriber interface on the MX Series and set an interval of 30 seconds for the keepalive value.

[edit interfaces pp0]
user@host# **set unit 3 keepalives interval 30**

b. Configure the underlying interface **ge-3/0/0.3** and PPPoE server mode for a dynamic PPPoE logical interface in a dynamic profile.

[edit interfaces pp0 unit 3]
user@host# **set pppoe-options underlying interface ge-3/0/0.3 server**

c. Configure PPP-specific interface properties in a dynamic profile: **pap** and **chap**, and set the **dynamic-profile** to the services dynamic profile.

NOTE: The dynamic profile is applied when Link Control Protocol (LCP) is negotiated in PPP.

[edit interfaces pp0 unit 3]
user@host# **set ppp-options pap chap dynamic-profile pp0-l2l3-service prof**

d. Configure the static bundle for the PPPoE member link for MLPPP subscribers using the **family mlppp** statement.

NOTE: The **family mlppp** statement determines whether MLPPP is supported for subscribers coming in from the underlying interface.

[edit interfaces pp0 unit 3]
user@host# **set family mlppp bundle si-1/0/0.3**

e. Configure tunneled subscribers.

[edit interfaces pp0 unit 3]
user@host# **set family inet**

7. If you are done configuring the device, commit the configuration.

[edit]
user@host# commit

Configuring a Static Inline Services MLPPP Bundle IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the static inline services (**si**) interface MLPPP bundle IFL, you specify the **encapsulation multilink-ppp** statement within the **si** interface. The **si** interface anchors the bundle interface.

You can also set these optional MLPPP parameters: MRRU, short sequence, and fragment-threshold. The following example shows how to configure the static **si** interface MLPPP bundle IFL:

1. Create the static (**si**) interface MLPPP bundle IFL **si-5/0/0** with a unit of 100.

   [edit interfaces]
   user@host# **set si-5/0/0 unit 100**

2. Configure the **encapsulation multilink-ppp** statement to enable MLPPP bundling for the **si-5/0/0.100** interface.

   [edit interfaces si-5/0/0.100]
   user@host# **set encapsulation multilink-ppp**

3. Configure the following MLPPP options for this example:

   - **mrru**—Specifies the maximum received reconstructed unit value ranging from 1500 through 4500 bytes.
   - **fragment-threshold**—Applies to all packets and forwarding classes, ranging from 128 through 16,320 bytes.
   - **short-sequence**—Determines the header format for the MLPPP. Default is **long-sequence**.

   [edit interfaces si-5/0/0.100]
   user@host# **set mrru 1500**
   user@host# **set fragment-threshold 640**
   user@host# **set short-sequence**

4. Enable support for static **si** interface IFL dynamic services by configuring the **ppp-options dynamic profile** statement.


```
[edit interfaces si-5/0/0.100]
user@host# set ppp-options dynamic-profile l2l3-service-prof
```

5. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces ge-3/0/0
  interfaces {
      ge-3/0/0 {
      vlan-tagging;
      unit 1 {
        encapsulation ppp-over-ether;
        vlan-id 1;
      }
      unit 2 {
        encapsulation ppp-over-ether;
        vlan-id 2;
      }
      unit 3 {
        encapsulation ppp-over-ether;
        vlan-id 3;
      }
    }
    pp0 {
      unit 1 {
        keepalives interval 30;
        pppoe-options {
          underlying-interface ge-3/0/0.1;
          server;
        }
        ppp-options {
          pap;
          chap;
          dynamic-profile pp0-l2l3-service-prof;
        }
        family mlppp {
          bundle si-1/0/0.1;
        }
        family inet {
          unnumbered-address lo0.0;
        }
        family inet6 {
          address 2040:2004::1.1.1.2/64;
        }
      }
      unit 2 {
        keepalives interval 30;
```

```
        pppoe-options {
          underlying-interface ge-3/0/0.2;
          server;
        }
        ppp-options {
          pap;
          dynamic-profile pp0-l2l3-service-prof;
        }
        family mlppp {
          bundle si-1/0/0.2;
        }
      }
      unit 3 {
        keepalives interval 30;
        pppoe-options {
          underlying-interface ge-3/0/0.3;
          server;
        }
        ppp-options {
          pap;
          chap;
          dynamic-profile pp0-l2l3-service-prof;
        }
        family mlppp {
          bundle si-1/0/0.3;
        }
        family inet;
      }
    }
  }

user@host# show interfaces si-5/1/0
  interfaces {
    si-5/1/0 {
      unit 100 {
        encapsulation multilink-ppp;
        mrru 1500;
        fragment-threshold 640;
        short-sequence;
        ppp-options {
          dynamic-profile l2l3-service-prof;
        }
      }
    }
  }
```

Verification

Confirm that the configuration is working properly.

- Verifying the Bundle IFL Information on page 63
- Verifying the Member Link IFL Information on page 65
- Verifying the Subscriber Information on page 66

Verifying the Bundle IFL Information

Purpose Verify that the bundle IFL information is correct for PPPoE MLPPP subscribers.

Action `root@haverhill>` **show interfaces si-5/1/0.1073756926 extensive**

```
Logical interface si-5/1/0.1073756926 (Index 102) (SNMP ifIndex 607)
(Generation 167)
 Flags: Up Point-To-Point SNMP-Traps 0x84000 Encapsulation: Multilink-PPP
 Last flapped: 2011-04-08 14:13:21 PDT (00:41:48 ago)
 Bandwidth: 10000mbps
 Bundle links information:
   Active bundle links       1
   Removed bundle links      0
   Disabled bundle links     0
 Bundle options:
   MRRU                                 1504
   Remote MRRU                          1504
   Drop timer period                    0
   Inner PPP Protocol field compression disabled
   Sequence number format               long (24 bits)
   Fragmentation threshold              500
   Links needed to sustain bundle       1
   Interleave fragments                 Enabled
   Multilink classes                    0
   Link layer overhead                  4.0 %
 Bundle status:
   Received sequence number        0xffffff
   Transmit sequence number        0xffffff
   Packet drops                    0 (0 bytes)
   Fragment drops                  0 (0 bytes)
   MRRU exceeded                   0
   Fragment timeout                0
   Missing sequence number         0
   Out-of-order sequence number    0
   Out-of-range sequence number    0
   Packet data buffer overflow     0
   Fragment data buffer overflow   0
 Statistics          Frames        fps         Bytes          bps
 Bundle:
   Multilink:
     Input :              3          0           270            0
     Output:              3          0           285            0
   Network:
     Input :              3          0           252            0
     Output:              3          0           276            0
 IPV6 Transit Statistics         Packets       Bytes
   Network:
     Input :                       0             0
     Output:                       0             0
 Link:
   pp0.1073756925
     Up time: 00:06:37
     Input :            126          0          9596            0
     Output:            126          0          1226            0
 Multilink detail statistics:
 Bundle:
   Fragments:
     Input :              0          0             0            0
     Output:              0          0             0            0
   Non-fragments:
     Input :              0          0             0            0
     Output:              0          0             0            0
   LFI:
```



```
       Input :             0          0             0            0
       Output:             0          0             0            0
 NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
   Protocol inet, MTU: 1500, Generation: 154, Route table: 0
     Flags: Sendbcast-pkt-to-re
     Addresses, Flags: Is-Primary
       Destination: Unspecified, Local: 80.80.80.1, Broadcast: Unspecified,
Generation: 150
```

Meaning Due to the particulars of implementation, the following error counts associated with a bundle always display 0: packet drops (bytes), fragment drops (bytes), fragment timeout, missing sequence number, out-of-order sequence number, out-of-range sequence number, packet data buffer overflow and fragment data buffer overflow, and MRRU exceeded.

Verifying the Member Link IFL Information

Purpose Verify that the member link IFL information is correct for subscribers.

Action `root@haverhill>` **show interfaces extensive pp0.1073756923**

```
  Logical interface pp0.1073756923 (Index 484) (SNMP ifIndex 708)
   (Generation 15544)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 38,
      Session AC name: haverhill, Remote MAC address: 00:00:64:24:01:02,
      Underlying interface: ge-1/0/0.50 (Index 423)
    Bandwidth: 1000mbps
    Traffic statistics:
     Input  bytes  :                  609
     Output bytes  :                  489
     Input  packets:                   21
     Output packets:                   22
    Local statistics:
     Input  bytes  :                  133
     Output bytes  :                  377
     Input  packets:                    7
     Output packets:                    8
    Transit statistics:
     Input  bytes  :                  476                    0 bps
     Output bytes  :                  112                    0 bps
     Input  packets:                   14                    0 pps
     Output packets:                   14                    0 pps
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  LCP state: Opened
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
mpls:
  Not-configured
  CHAP state: Success
  PAP state: Closed
    Protocol mlppp, Multilink bundle: si-1/0/0.1073756924
    Service interface: si-1/0/0, Dynamic profile: ml-bundle-service-prof
    MTU: 1526, Generation: 15535, Route table: 0
```

Meaning Logical interface **pp0.1073756923** has been configured with PPPoE, multilink bundle **si-1/0/0.1073756924**, and protocol **mlppp**.

Verifying the Subscriber Information

Purpose Verify that the subscriber information for static MLPPP over PPPoE is correct.

Action

```
root@haverhill> show subscribers detail
Type: PPPoE
User Name: user
IP Address: 10.4.1.2
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: pp0.20
Interface type: Static
MAC Address: 00:00:64:04:01:02
State: Active
Radius Accounting ID: 4
Session ID: 4
Bundle Session ID: 5
Login Time: 2012-02-28 10:32:24 PST

Type: MLPPP
User Name: user
IP Address: 10.4.1.2
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: si-1/0/0.1020
Interface type: Static
State: Active
Radius Accounting ID: 5
Session ID: 5
Underlying Session ID: 4
Login Time: 2012-02-28 10:32:24 PST
```

Meaning

Subscriber information has been configured for static PPPoE with interface **pp0.20**, and static MLPPP with interface **si-1/0/0.1020**.

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11
- Example: Configuring Dynamic PPPoE MLPPP Subscribers on page 87

CHAPTER 10

Dynamic MLPPP Subscribers Configuration for MX Series Examples

Example: Configuring Dynamic LNS MLPPP Subscribers

This example shows how to configure dynamic L2TP network server (LNS) multilink (MLPPP) subscribers.

Requirements

This example uses the following hardware and software components:

- MX Series with MPC2s installed
- Junos OS Release 13.3 or later

Before you configure dynamic LNS MLPPP subscribers, be sure you have:

- If configuring a tunnel group using an inline service (**si**) interface, enabled the inline service (**si**) interface for LNS subscribers. See "Enabling Inline Service Interfaces for PPPoE and LNS Subscribers" on page 34.
- Configured the inline service (**si**) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 35.
- If configuring a tunnel group using a pool of service interfaces, configured service device pools for LNS subscribers. See "Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers" on page 36.

Overview

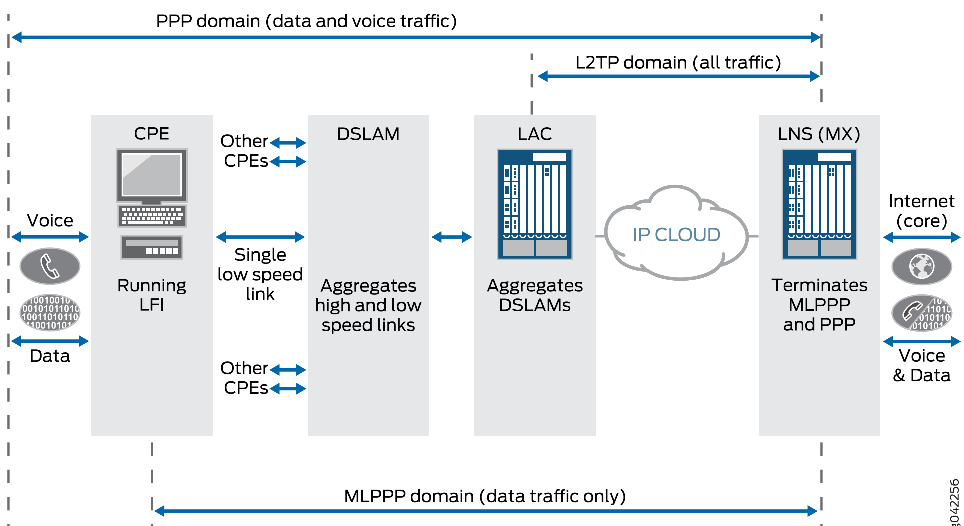
An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For dynamic LNS MLPPP subscribers, you configure the dynamic member link IFLs using dynamic profiles. The member link dynamic profile includes the **family mlppp** statement containing the bundle dynamic profile and the service interface (**si**), or a pool of service interfaces. This information is then used to create the dynamic bundle IFL.

Each dynamic bundle accepts only one dynamic member link. If more than one dynamic member link attempts to join the same dynamic bundle, the system fails the new member session.

Figure 16 on page 70 shows how the different types of traffic traverse through a network where the MX Series is acting as the LNS to terminate MLPPP bundles.

Topology

Figure 16: MLPPP Bundles Terminated at MX Series as the LNS Network



The following three domains are shown passing traffic through the LNS network:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only
- L2TP domain—Contains all types of traffic

Configuration

To configure dynamic LNS MLPPP subscribers, perform these tasks:

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.

```
[edit]
set access profile ce-l2tp-profile2 client ce-lac-3 user-group-profile ce-lac-1-gp
set access profile ce-l2tp-profile2 client ce-lac-3 l2tp multilink
set access profile ce-l2tp-profile2 client ce-lac-3 l2tp maximum-sessions-per-tunnel
  2000
set access profile ce-l2tp-profile2 client ce-lac-3 l2tp shared-secret “password”
set access profile ce-l2tp-profile2 client ce-lac-3 l2tp dynamic-profile ml-lns-member-prof
set services l2tp tunnel-group dyn-l2tp-tunnel-group l2tp-access-profile ce-l2tp-profile2
set services l2tp tunnel-group dyn-l2tp-tunnel-group aaa-access-profile ce-authenticator
set services l2tp tunnel-group dyn-l2tp-tunnel-group local-gateway address 31.1.1.1
set services l2tp tunnel-group dyn-l2tp-tunnel-group service-device-pool pool1
set services l2tp tunnel-group dyn-l2tp-tunnel-group dynamic-profile ml-lns-member-prof

[edit]
set dynamic-profiles mlp-lns-member-profile
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit”
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit” dial-options l2tp-interface-id dont care dedicated
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit” family mlppp bundle “$junos-bundle-interface-name”
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit” family mlppp service-device-pool pool1
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit” family mlppp dynamic-profile ml-bundle-prof
set dynamic-profiles mlp-lns-member-profile interfaces “$junos-interface-ifd-name”
  unit “$junos-interface-unit” family inet

[edit]
set dynamic-profiles ml-bundle-prof
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name"
```

```
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix next-hop $junos-framed-route-nexthop
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix metric $junos-framed-route-cost
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix preference $junos-framed-route-distance
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" access-internal route $junos-subscriber-ip-address
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" access-internal route $junos-subscriber-ip-address
  qualified-next-hop $junos-interface-name
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit”
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” encapsulation multilink-ppp
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” mrru 1500
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” short-sequence
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” fragment-threshold 320
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family inet
set class-of-service traffic-control-profiles tcp2
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  scheduler-map "$junos-cos-scheduler-map”
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  shaping-rate "$junos-cos-shaping-rate"
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  guaranteed-rate "$junos-cos-guaranteed-rate"
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  delay-buffer-rate “$junos-cos-delay-buffer-rate”
set dynamic-profiles ml-bundle-prof class-of-service interfaces
  “$junos-interface-ifd-name” unit "$junos-interface-unit"
set dynamic-profiles ml-bundle-prof class-of-service interfaces
  “$junos-interface-ifd-name” unit "$junos-interface-unit" output-traffic-control-profile
  tcp2
set dynamic-profiles ml-bundle-prof class-of-service interfaces
  “$junos-interface-ifd-name” unit "$junos-interface-unit" fragmentation-map fragmap-2
```


Configuring a Tunnel Group with a Pool of Service Interfaces and L2TP Access Profile Attributes

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure a tunnel group with a pool of service interfaces and L2TP access profile attributes for dynamic LNS MLPPP subscribers:

1. Create the access profile.

   [edit access]
   user@host# **set profile ce-l2tp-profile2**

2. Configure an L2TP (LAC) access client.

   [edit access profile ce-l2tp-profile2]
   user@host# **set client ce-lac-3**

3. Associate a group profile containing PPP attributes to apply for the PPP sessions being tunneled from this LAC client.

   [edit access profile ce-l2tp-profile2 client ce-lac-3]
   user@host# **set user-group-profile ce-lac-1-gp**

4. Configure the following L2TP access profile attributes for this example:

   - L2TP client is multilink (MLPPP)-capable for subscribers. The **multilink** statement in the L2TP access client profile determines whether MLPPP is supported for subscribers coming in from the LAC peer.
   - Maximum number of sessions allowed in a tunnel from the client (LAC).
   - Tunnel password used to authenticate the client (LAC).
   - Dynamic profile name in the L2TP access client profile for dynamic LNS MLPPP subscribers.

   NOTE: If the **dynamic-profile** ***name*** is defined in the L2TP access client profile, it is used to create the dynamic LNS MLPPP member link; otherwise, the **dynamic-profile** ***name*** defined in the tunnel group is used. If neither profile contains the **family mlppp** statement, then the incoming LNS session fails.

   [edit access profile ce-l2tp-profile2 client ce-lac-3]
   user@host# **set l2tp multilink**
   user@host# **set l2tp maximum-sessions-per-tunnel 2000**
   user@host# **set l2tp shared-secret *password***
   user@host# **set dynamic-profile ml-lns-member-prof**

5. Create the tunnel group.

   [edit services l2tp]

user@host# **set tunnel-group dyn-l2tp-tunnel-group**

6. Set the tunnel access profile equal to the setting you defined for the access profile.

   [edit services l2tp tunnel-group dyn-l2tp-tunnel-group]
   user@host# **set l2tp-access-profile ce-l2tp-profile2**

7. Set the L2TP AAA access profile.

   NOTE: You can specify the L2TP AAA access profile in either the **[edit access]** or **[edit services]** hierarchy levels, using the LNS access client profile or **tunnel-group** statements, respectively. An L2TP AAA access profile defined using the **[edit access]** hierarchy level overrides the L2TP AAA access profile defined for the **tunnel-group** using the **[edit services]** hierarchy level.

   [edit services l2tp tunnel-group dyn-l2tp-tunnel-group]
   user@host# **set aaa-access-profile ce-authenticator**

8. Set the local gateway address for the L2TP tunnel.

   [edit services l2tp tunnel-group dyn-l2tp-tunnel-group]
   user@host# **set local-gateway address 31.1.1.1**

9. Specify the pool of service interfaces for the dynamic LNS MLPPP subscribers.

   [edit services l2tp tunnel-group dyn-l2tp-tunnel-group]
   user@host# **set service-device-pool pool1**

10. Specify the dynamic profile used to create the dynamic LNS MLPPP member link.

    [edit services l2tp tunnel-group dyn-l2tp-tunnel-group]
    user@host# **set dynamic-profile ml-lns-member-prof**

11. If you are done configuring the device, commit the configuration.

    [edit]
    user@host# **commit**

Configuring a Dynamic Profile for Dynamic LNS Member Link IFL Without Mixed Mode Support

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

You can configure the **dynamic-profile *name*** used to create the dynamic LNS member link IFL in either the L2TP client access profile or in the tunnel-group. See "Configuring a Tunnel Group with a Pool of Service Interfaces and L2TP Access Profile Attributes" on page 73.

The following example shows **dynamic-profile** configuration for LNS MLPPP and PPP subscribers. The **family mlppp** statement contains the **dynamic-profile *name***, and either the **service-interface** or the **service-device-pool**, used to create the dynamic bundle IFL. If you configure a **service-device-pool**, an inline services (**si**) interface is selected from the pool to create the dynamic bundle IFL using a round-robin method.

You must also configure the **family inet** statement in the **si** member link dynamic profile interface for tunneled subscribers. The **family inet** statement enables the L2TP long route to be installed and supported for the lookup engine to steer control packets to the Routing Engine.

NOTE: Optionally, you can configure the dynamic profile to support mixed mode to enable PPP subscribers to successfully log in using the dynamic profile. See "Configuring a Dynamic Profile for Dynamic LNS Member Link IFL With Mixed Mode Support" on page 77 for the additional configuration commands required.

1. Specify the dynamic profile that you used to create the dynamic LNS MLPPP member link previously in "Configuring a Tunnel Group with a Pool of Service Interfaces and L2TP Access Profile Attributes" on page 73.

   [edit dynamic-profiles]
   user@host# **set ml-lns-member-prof**

2. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

NOTE: The interface setting for a dynamic profile for PPPoE sessions can use either of the following code formats:

- **set interfaces pp0**

  or

- **set interfaces "$junos-interface-ifd-name"**

This example uses **set interfaces "$junos-interface-ifd-name"**.

[edit dynamic-profiles ml-lns-member-prof]
user@host# **set interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"**

3. For the *$junos-interface-ifd-name* interface, set the L2TP interface dial options to specify that the logical interface can host one session at a time (dedicated).

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set dial-options l2tp-interface-id dont care dedicated**

4. Enable MLPPP support for LNS MLPPP subscribers and configure the dynamic bundle interface (IFL) by setting the predefined dynamic bundle interface variable *$junos-bundle-interface-name*.

   NOTE: The **family mlppp** statement determines whether MLPPP is supported for subscribers coming in from the underlying interface.

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set family mlppp bundle "$junos-bundle-interface-name"**

5. Specify the pool of service interfaces for the dynamic LNS MLPPP subscribers.

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" family mlppp]
   user@host# **set service-device-pool pool1**

6. Specify the dynamic profile name for the bundle.

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" family mlppp]
   user@host# **set dynamic-profile ml-bundle-prof**

7. Enable support for LNS subscribers and the LNS long route.

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set family inet**

8. If you are done configuring the device, commit the configuration.

[edit]
user@host# **commit**

Configuring a Dynamic Profile for Dynamic LNS Member Link IFL With Mixed Mode Support

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

Optionally, you can configure the dynamic profile to support mixed mode to enable PPP subscribers to successfully log in using the dynamic profile.

The following example shows the additional configurations required to support mixed mode for dynamic profiles.

NOTE: The following configuration commands are not included in the “CLI Quick Configuration” on page 71 section.

1. Specify the dynamic profile that you used to create the dynamic LNS MLPPP member link previously in “Configuring a Tunnel Group with a Pool of Service Interfaces and L2TP Access Profile Attributes” on page 73.

   [edit dynamic-profiles]
   user@host# **set ml-lns-member-prof**

2. When the customer premises equipment (CPE) is for a dynamic virtual routing and forwarding (VRF) PPP subscriber, you must configure the routing instance and its interface.

   [edit dynamic-profiles ml-lns-member-prof]
   user@host# **set routing-instances "$junos-routing-instance" interface "$junos-interface-name"**

3. Configure the access route for the routing options.

   [edit dynamic-profiles ml-lns-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access route $junos-framed-route-ip-address-prefix**

4. Configure the next-hop, metric, and preference for the router.

   [edit dynamic-profiles ml-lns-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access route $junos-framed-route-ip-address-prefix]
   user@host# **set next-hop $junos-framed-route-nexthop**
   user@host# **set metric $junos-framed-route-cost**
   user@host# **set preference $junos-framed-route-distance**

5. Configure the internal access route for the routing options.

   [edit dynamic-profiles ml-lns-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]

user@host# **set routing-options access-internal route $junos-subscriber-ip-address**

6. Configure the qualified next-hop for the internal route..

   [edit dynamic-profiles ml-lns-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access-internal route $junos-subscriber-ip-address]
   user@host# **set qualified-next-hop $junos-interface-name**

7. Follow the procedure described in "Configuring a Dynamic Profile for Dynamic LNS Member Link IFL Without Mixed Mode Support" on page 75 to configure the basic settings for the dynamic profile.

   NOTE: To enable mixed mode support, when the CPE is a PPP subscriber, you must also add an unnumbered address, and input and output filters to the **family inet** statement.

   [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set family inet unnumbered-address $junos-loopback-interface**
   user@host# **set family inet filter input "$junos-input-filter" output "$junos-output-filter"**

8. When the CPE is a PPP subscriber, you must also configure class of service and define the traffic control profile.

   [edit dynamic-profiles ml-lns-member-prof class-of-service]
   user@host# **set traffic-control-profiles tc-profile**

9. For the traffic-control profile, define the following settings: scheduler map, shaping rate, overhead accounting, guaranteed rate, and delay buffer rate.

   [edit dynamic-profiles ml-lns-member-prof 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 "$junos-cos-shaping-mode" bytes "$junos-cos-byte-adjust"**
   user@host# **set guaranteed-rate "$junos-cos-guaranteed-rate"**
   user@host# **set delay-buffer-rate "$junos-cos-delay-buffer-rate"**

10. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*.

    [edit dynamic-profiles ml-lns-member-prof class-of-service]
    user@host# **set interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"**

11. For the dynamic profile interface, define the following settings: output traffic control profile, classifiers, and rewrite rules.

[edit dynamic-profiles ml-lns-member-prof class-of-service interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
user@host# **set output-traffic-control-profile tc-profile**
user@host# **set classifiers dscp GEN-CLASSIFIER-IN**
user@host# **set rewrite-rules dscp GEN-RW-OUT-DSCP**

12. If you are done configuring the device, commit the configuration.

    [edit]
    user@host# **commit**

Configuring a Dynamic Profile for the Dynamic Bundle IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the dynamic profile for the dynamic bundle IFL, you specify the **encapsulation multilink-ppp** statement within the dynamic profile. The **dynamic profile** for the dynamic bundle IFL is referenced from the **dynamic profile** for dynamic PPPoE and LNS member link IFLs.

You must configure the **fragmentation-maps** statement statically using class-of-service and assign them in the bundle dynamic profile. You can also set these optional MLPPP parameters: MRRU, short sequence, and fragment-threshold. The following example shows how to configure the dynamic profile for the dynamic bundle IFL.

1. Specify the dynamic profile name for the bundle.

   [edit dynamic-profiles}
   user@host# **set ml-bundle-prof**

2. Although MLPPP member links process authentication and routing-instance assignments, if a non-default routing-instance is assigned, you must configure the bundle IFL under the assigned routing-instance. As a result, you must also configure routing-instances in the bundle dynamic-profile.

   [edit dynamic-profiles ml-bundle-prof]
   user@host# **set routing-instances "$junos-routing-instance" interface "$junos-interface-name"**

3. Configure the access route for the routing options.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access route $junos-framed-route-ip-address-prefix**

4. Configure the next-hop, metric, and preference for the router.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access route $junos-framed-route-ip-address-prefix]
   user@host# **set next-hop $junos-framed-route-nexthop**
   user@host# **set metric $junos-framed-route-cost**
   user@host# **set preference $junos-framed-route-distance**

5. Configure the internal access route for the routing options.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access-internal route $junos-subscriber-ip-address**

6. Configure the qualified next-hop for the internal route.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access-internal route $junos-subscriber-ip-address]
   user@host# **set qualified-next-hop $junos-interface-name**

7. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

   [edit dynamic-profiles ml-bundle-prof]
   user@host# **set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”**

8. Configure the **encapsulation multilink-ppp** statement to enable MLPPP bundling for the dynamic profile.

   [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
   user@host# **set encapsulation multilink-ppp**

9. Configure the following MLPPP options for this example:

   - **mrru**—Specifies the maximum received reconstructed unit value ranging from 1500 through 4500 bytes.
   - **fragment-threshold**—Applies to all packets and forwarding classes, ranging from 128 through 16,320 bytes.
   - **short-sequence**—Determines the header format for the MLPPP. Default is **long-sequence**.

   [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
   user@host# **set mrru 1500**
   user@host# **set fragment-threshold 320**
   user@host# **set short-sequence**

10. Enable support for MLPP subscribers.

    [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
    user@host# **set family inet**

11. To enable **fragmentation-maps** support, you must configure class-of-service and define the traffic control profile.

    [edit dynamic-profiles ml-bundle-prof class-of-service]
    user@host# **set traffic-control-profiles tcp2**

12. For the traffic-control profile, define the following settings: scheduler map, shaping rate, guaranteed rate, and delay buffer rate.

    ```
    [edit dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2]
    user@host# set scheduler-map "$junos-cos-scheduler-map"
    user@host# set shaping-rate "$junos-cos-shaping-rate"
    user@host# set guaranteed-rate "$junos-cos-guaranteed-rate"
    user@host# set delay-buffer-rate "$junos-cos-delay-buffer-rate"
    ```

13. Configure the underlying interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

    ```
    [edit dynamic-profiles ml-bundle-prof class-of-service]
    user@host# set interfaces “$junos-interface-ifd-name” unit "$junos-interface-unit"
    ```

14. For the dynamic profile interface, define the output traffic control profile.

    ```
    [edit dynamic-profiles ml-bundle-prof class-of-service interfaces
    “$junos-interface-ifd-name” unit "$junos-interface-unit"]
    user@host# set output-traffic-control-profile tcp2
    ```

15. Define the fragmentation-map required for dynamic profile bundles and used to enable link fragmentation and interleaving (LFI).

    ```
    [edit dynamic-profiles ml-bundle-prof class-of-service interfaces
    “$junos-interface-ifd-name” unit "$junos-interface-unit"]
    user@host# set fragmentation-map fragmap-2
    ```

16. If you are done configuring the device, commit the configuration.

    ```
    [edit]
    user@host# commit
    ```

Results

From configuration mode, confirm your configuration by entering the **show access**, **show services**, and **show dynamic-profiles** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show access profile ce-l2tp-profile2
  access profile {
     ce-l2tp-profile2 {
       client ce-lac-3 {
          user-group-profile ce-lac-1-gp;
          l2tp {
             multilink;
             interface-id not-used;
             maximum-sessions-per-tunnel 2000;
             shared-secret "$9$2wgUHQF/9pB";
             dynamic-profile ml-lns-member-prof;
          }
       }
```

```
    }
  }
user@host# show services l2tp tunnel-group dyn-l2tp-tunnel-group
  services {
    l2tp {
      tunnel-group dyn-l2tp-tunnel-group {
        l2tp-access-profile ce-l2tp-profile2;
        aaa-access-profile ce-authenticator;
        local-gateway {
          address 31.1.1.1;
        }
        service-device-pool pool1;
        dynamic-profile ml-lns-member-prof;
      }
    }
  }
```

Dynamic profile for dynamic LNS member link IFL without mixed mode:

```
user@host# show dynamic-profiles mlp-lns-member-profile
  dynamic-profile mlp-lns-member-profile {
    interface $junos-interface-ifd-name {
      unit $junos-interface-unit" {
        dial-options {
          l2tp-interface-id dont-care;
          dedicated;
        }
        family mlppp {
          bundle $junos-bundle-interface-name ;
          service-device-pool pool1;
          dynamic-profile mlp-bundle-profile;
        }
        family inet {
        }
      }
    }
  }
```

Dynamic profile for dynamic LNS member link IFL with mixed mode:

```
user@host# show dynamic-profiles mlp-lns-member-profile
  dynamic-profile ml-lns-member-prof {
   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 mlppp {
          bundle $junos-bundle-interface-name;
          service-device-pool pool2;
          dynamic-profile ml-bundle-prof;
        }
        family inet {
          unnumbered-address $junos-loopback-interface;
          filter {
            input "$junos-input-filter";
            output "$junos-output-filter";
          }
        }
      }
  class-of-service {
      traffic-control-profiles {
        tc-profile {
          scheduler-map "$junos-cos-scheduler-map";
          shaping-rate "$junos-cos-shaping-rate";
        overhead-accounting "$junos-cos-shaping-mode" bytes "$junos-cos-byte-adjust";

          guaranteed-rate "$junos-cos-guaranteed-rate";
          delay-buffer-rate "$junos-cos-delay-buffer-rate";
        }
      }
      interfaces {
        "$junos-interface-ifd-name" {
          unit "$junos-interface-unit" {
            output-traffic-control-profile tc-profile;
            classifiers {
              dscp GEN-CLASSIFIER-IN;
            }
            rewrite-rules {
              dscp GEN-RW-OUT-DSCP;
            }
          }
        }
      }
    }
  }

user@host# show dynamic-profiles ml-bundle-prof
  dynamic-profile ml-bundle-prof {
      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 "$junos-interface-unit" {
          encapsulation multilink_ppp;
          mrru 1500;
          short-sequence;
          fragment-threshold 320;
          family inet
          }
        }
      }
    class-of-service {
        traffic-control-profiles {
          tcp2 {
            scheduler-map "$junos-cos-scheduler-map";
            shaping-rate "$junos-cos-shaping-rate";
            guaranteed-rate "$junos-cos-guaranteed-rate";
            delay-buffer-rate "$junos-cos-delay-buffer-rate";
          }
        }
        interfaces {
          "$junos-interface-ifd-name" {
            unit "$junos-interface-unit" {
              output-traffic-control-profile tcp2;
              fragmentation-map fragmap-2
              }
           }
        }
      }
```

Verification

Confirm that the configuration is working properly.

- Verifying the Subscriber Information on page 85
- Verifying Mixed Mode Support with a Dynamic MLPPP-Capable Subscriber on page 85
- Verifying Tunneled MLPPP Over LAC Interfaces on page 86

Verifying the Subscriber Information

Purpose Verify that the subscriber information for dynamic MLPPP over LNS is correct.

Action

```
root@haverhill> show subscribers extensive
Type: L2TP
User Name: lns-client
IP Address: 80.80.80.20
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: si-1/0/0.1073741824
Interface type: Dynamic
Dynamic Profile Name: ml-lns-member-prof
Dynamic Profile Version: 1
State: Active
Radius Accounting ID: 20
Session ID: 20
Bundle Session ID: 21
Login Time: 2011-04-11 10:55:13 PDT

Type: MLPPP
User Name: lns-client
IP Address: 80.80.80.20
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: si-3/0/0.1073741825
Interface type: Dynamic
Dynamic Profile Name: ml-bundle-prof
Dynamic Profile Version: 1
State: Active
Radius Accounting ID: 21
Session ID: 21
Underlying Session ID: 20
Login Time: 2011-04-11 07:55:59 PDT
```

Meaning Subscriber information for interface **si-1/0/0.1073741824** has been configured for MLPPP with interface type of dynamic.

Verifying Mixed Mode Support with a Dynamic MLPPP-Capable Subscriber

Purpose Verify that mixed mode interfaces negotiated correctly for the single link PPP using a dynamic MLPPP-capable subscriber.

Action

```
root@haverhill> show interfaces extensive pp0.1073741832
Logical interface pp0.1073741832 (Index 489) (SNMP ifIndex 712)
   (Generation 299)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 40,
      Session AC name: haverhill1, Remote MAC address: 00:00:64:3c:01:02,
      Underlying interface: ge-1/0/0.44 (Index 376)
    Traffic statistics:
     Input  bytes  :              1213
     Output bytes  :              1672
     Input  packets:                41
     Output packets:                49
     IPv6 transit statistics:
      Input  bytes  :                0
      Output bytes  :                0
      Input  packets:                0
      Output packets:                0
    Local statistics:
     Input  bytes  :               159
     Output bytes  :              1424
     Input  packets:                10
     Output packets:                18
    Transit statistics:
     Input  bytes  :              1054                    0 bps
     Output bytes  :               248                    0 bps
     Input  packets:                31                    0 pps
     Output packets:                31                    0 pps
     IPv6 transit statistics:
      Input  bytes  :                0
      Output bytes  :                0
      Input  packets:                0
      Output packets:                0
  Keepalive settings: Interval 45 seconds, Up-count 1, Down-count 3
  LCP state: Opened
  NCP state: inet: Opened, inet6: Opened, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Success
    Protocol inet, MTU: 65531, Generation: 384, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Primary
        Destination: Unspecified, Local: 101.0.0.1, Broadcast: Unspecified,
        Generation: 297
    Protocol inet6, MTU: 65531, Generation: 385, Route table: 0
      Addresses, Flags: Is-Primary
        Destination: Unspecified, Local: 2030::1
    Generation: 298
        Destination: Unspecified, Local: fe80::2a0:a50f:fc64:6ef2
    Generation: 299
```

Meaning

When a dynamic MLPPP-capable subscriber negotiates a single link PPP, the results are the same as a non-MLPPP subscriber; no bundle IFL or SDB session is created.

Verifying Tunneled MLPPP Over LAC Interfaces

Purpose

Verify that the MLPPP over LAC member link IFL is correct.

Action

```
root@haverhill> show interfaces extensive pp0.1073756921
Logical interface pp0.1073756921 (Index 482) (SNMP ifIndex 706)
   (Generation 15542)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 37,
      Session AC name: haverhill, Remote MAC address: 00:00:64:1d:01:02,
      Underlying interface: ge-1/0/0.2040 (Index 457)
    Traffic statistics:
     Input  bytes  :                 273
     Output bytes  :                 270
     Input  packets:                  13
     Output packets:                  10
    Local statistics:
     Input  bytes  :                 138
     Output bytes  :                 155
     Input  packets:                   6
     Output packets:                   3
    Transit statistics:
     Input  bytes  :                 135                   0 bps
     Output bytes  :                 115                   0 bps
     Input  packets:                   7                   0 pps
     Output packets:                   7                   0 pps
  Keepalive settings: Interval 45 seconds, Up-count 1, Down-count 3
  LCP state: Opened
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 1492, Generation: 15534, Route table: 0
      Flags: Sendbcast-pkt-to-re
    Protocol mlppp, Multilink bundle: si-1/0/0.1073756922
    Service device pool: sipool-1, Dynamic profile: ml-bundle-prof
    MTU: 1526, Generation: 15533, Route table: 0
```

Meaning

When a PPPoE MLPPP session is tunneled, the bundle and member link binding remains. Although the bundle IFL does not participate in the control and forwarding path, it remains in the user-interface.

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Mixed Mode Support for MLPPP and PPP Subscribers Overview on page 9
- Configuring L2TP Client Access to Support MLPPP for Dynamic Subscribers on page 41

Example: Configuring Dynamic PPPoE MLPPP Subscribers

This example shows how to configure dynamic Point-to-Point Protocol over Ethernet (PPPoE) multilink (MLPPP) subscribers.

- Configuration on page 89
- Verification on page 100

Requirements

This example uses the following hardware and software components:

- MX Series with MPC2s installed
- Junos OS Release 13.3 or later

Before you configure dynamic PPPoE MLPPP subscribers, be sure you have:

- If configuring a tunnel group using an inline service (**si**) interface, enabled the inline service (**si**) interface for PPPoE subscribers. See "Enabling Inline Service Interfaces for PPPoE and LNS Subscribers" on page 34.
- Configured the inline service (**si**) interface for PPPoE subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 35.
- If configuring a tunnel group using a pool of service interfaces, configured service device pools for PPPoE subscribers. See "Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers" on page 36.

Overview

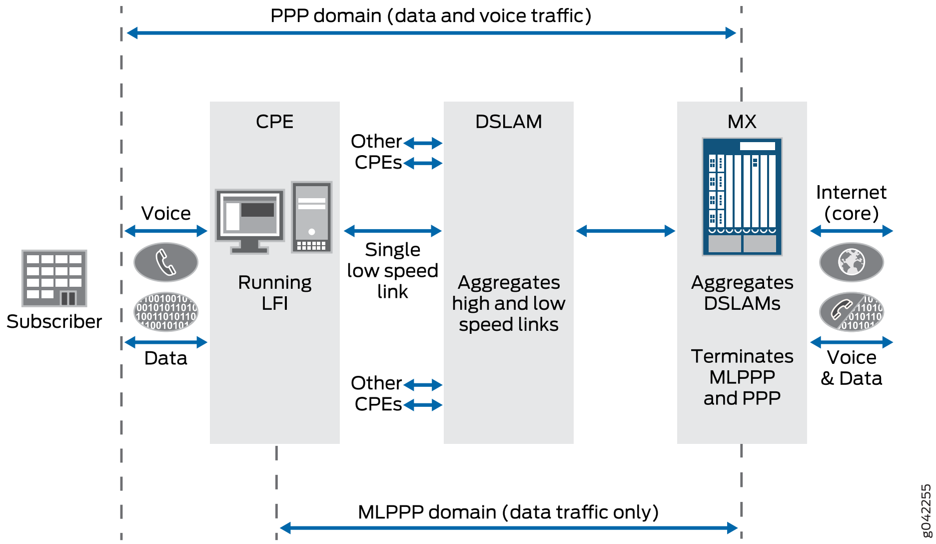
An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For dynamic PPPoE MLPPP subscribers, you configure the dynamic **pp0** member link IFLs using dynamic profiles. The **pp0** member link dynamic profile includes the **family mlppp** statement containing the dynamic profile name and the service interface (**si**), or a pool of service interfaces. This information is then used to create the dynamic bundle IFL.

Each dynamic bundle accepts only one dynamic member link. If more than one dynamic member link attempts to join the same dynamic bundle, the system fails the new member session.

Figure 17 on page 89 shows how the different types of traffic traverse through a network where the MX Series terminates PPPoE sessions.

Topology

Figure 17: PPP and MLPPP Traffic Terminated at MX Series



The following two domains are shown terminating traffic at the MX Series:

- PPP domain—Contains data and voice traffic
- MLPPP domain—Contains data traffic only

Configuration

To configure dynamic PPPoE MLPPP subscribers, perform these tasks:

- Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL Without Mixed Mode Support on page 91
- Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL With Mixed Mode Support on page 93
- Configuring a Dynamic Profile for the Dynamic Bundle IFL on page 96
- Results on page 98

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.

```
[edit]
set interfaces ge-1/0/0 flexible-vlan-tagging
set interfaces ge-1/0/0 unit 600 encapsulation ppp-over-ether vlan-id 600
set interfaces ge-1/0/0 unit 600 pppoe-underlying-options dynamic-profile
  ml-pp0-member-prof
set dynamic-profiles ml-pp0-member-prof
```

```
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit”
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” pppoe-options underlying-interface
  "$junos-underlying-interface" server
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” ppp-options pap chap lcp-restart-timer 5000
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family mlppp bundle “$junos-bundle-interface-name”
set dynamic-profiles ml-pp0-member-profinterfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family mlppp service-interface si-5/1/0
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family mlppp dynamic-profile ml-bundle-prof
set dynamic-profiles ml-pp0-member-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family inet

[edit]
set dynamic-profiles ml-bundle-prof
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name"
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix next-hop $junos-framed-route-nexthop
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix metric $junos-framed-route-cost
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" routing-options access route
  $junos-framed-route-ip-address-prefix preference $junos-framed-route-distance
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" access-internal route $junos-subscriber-ip-address
set dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface
  "$junos-interface-name" access-internal route $junos-subscriber-ip-address
  qualified-next-hop $junos-interface-name
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit”
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” encapsulation multilink-ppp
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” mrru 1500
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” short-sequence
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” fragment-threshold 320
set dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit
  “$junos-interface-unit” family inet
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  scheduler-map "$junos-cos-scheduler-map”
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2
  shaping-rate "$junos-cos-shaping-rate"
```


set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2 guaranteed-rate "$junos-cos-guaranteed-rate"
set dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2 delay-buffer-rate "$junos-cos-delay-buffer-rate"
set dynamic-profiles ml-bundle-prof class-of-service interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"
set dynamic-profiles ml-bundle-prof class-of-service interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" output-traffic-control-profile tcp2
set dynamic-profiles ml-bundle-prof class-of-service interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" fragmentation-map fragmap-2

Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL Without Mixed Mode Support

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

You configure the dynamic **pp0** member link IFLs by using dynamic profiles and including the **family mlppp** statement. The **family mlppp** statement contains the **dynamic-profile *name***, and either the **service-interface** or the **service-device-pool** used to create the dynamic bundle IFL. If you configure a **service-device-pool**, an inline services (**si**) interface is selected from the pool to create the dynamic bundle IFL using a round-robin method.

You must also configure the **family inet** statement in the tunneled **pp0** member link dynamic profile. The **family inet** statement enables the L2TP long route to be installed and supported for the lookup engine to steer control packets to the Routing Engine.

NOTE: Optionally, you can configure the dynamic profile to support mixed mode to enable PPP subscribers to successfully log in using the dynamic profile. See "Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL With Mixed Mode Support" on page 93 for the additional configuration commands required.

The following example shows how to configure dynamic **pp0** member link IFLs over flexible VLAN to support PPPoE MLPPP subscribers.

1. Create the Gigabit Ethernet underlying interface for the dynamic profile, **ge-1/0/0** and enable flexible VLAN tagging.

   [edit interfaces]
   user@host# **set ge-1/0/0 flexible vlan-tagging**

2. For the **ge-1/0/0** interface, configure PPP over Ethernet encapsulation for VLAN 600.

   [edit interfaces ge-1/0/0]
   user@host# **set unit 600 encapsulation ppp-over-ether vlan-id 600**

3. Configure the PPPoE underlying interface and set its dynamic profile.

   [edit interfaces ge-1/0/0 unit 600]

user@host# **set pppoe-underlying-options dynamic-profile ml-pp0-member-prof**

4. Specify the dynamic profile that you previously set as the PPPoE underlying interface dynamic profile.

   [edit dynamic-profiles]
   user@host# **set ml-pp0-member-prof**

5. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

   NOTE: The interface setting for a dynamic profile for PPPoE sessions can use either of the following code formats:

   - **set interfaces pp0**

     or

   - **set interfaces "$junos-interface-ifd-name"**

   This example uses **set interfaces "$junos-interface-ifd-name"**.

   [edit dynamic-profiles ml-pp0-member-prof]
   user@host# **set interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"**

6. For the *$junos-interface-ifd-name* interface, configure the underlying interface for the PPPoE options and PPPoE server mode for a dynamic PPPoE logical interface in a dynamic profile.

   [edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set pppoe-options underlying-interface "$junos-underlying-interface" server**

7. Configure PPP-specific interface properties in a dynamic profile: **pap**, **chap**, and set the **lcp-restart-timer** to 5000.

   [edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set ppp-options pap chap lcp-restart-timer 5000**

8. Enable MLPPP support for dynamic PPPoE MLPPP subscribers and configure the dynamic bundle interface (IFL) by setting the predefined dynamic bundle interface variable *$junos-bundle-interface-name*.

   NOTE: The **family mlppp** statement determines whether MLPPP is supported for subscribers coming in from the subscriber interface.

[edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# **set family mlppp bundle "$junos-bundle-interface-name"**

9. Specify the service interface for the dynamic PPPoE MLPPP subscribers.

   [edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" family mlppp]
   user@host# **set service-interface si-5/1/0**

10. Specify the dynamic profile name for the bundle.

    [edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit" family mlppp]
    user@host# **set dynamic-profile ml-bundle-prof**

11. Enable support for PPPoE tunneled subscribers and the LAC long route.

    [edit dynamic-profiles ml-lns-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
    user@host# **set family inet**

12. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL With Mixed Mode Support

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

Optionally, you can configure the dynamic profile to support mixed mode to enable PPP subscribers to successfully log in using the dynamic profile.

The following example shows the additional configurations required to support mixed mode for dynamic profiles.

NOTE: **The following configuration commands are not included in the "CLI Quick Configuration" on page 89 section.**

1. Configure dynamic **pp0** member link IFLs over flexible VLAN to support PPPoE MLPPP subscribers. See "Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL Without Mixed Mode Support" on page 91, steps 1 through 4.
2. Specify the dynamic profile that you used to create the dynamic PPPoE MLPPP member link.

   [edit dynamic-profiles]
   user@host# **set ml-pp0-member-prof**

3. When the customer premises equipment (CPE) is for a dynamic virtual routing and forwarding (VRF) PPP subscriber, you must configure the routing instance and its interface.

   [edit dynamic-profiles ml-pp0-member-prof]
   user@host# **set routing-instances "$junos-routing-instance" interface "$junos-interface-name"**

4. Configure the access route for the routing options.

   [edit dynamic-profiles ml-pp0-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access route $junos-framed-route-ip-address-prefix**

5. Configure the next-hop, metric, and preference for the router.

   [edit dynamic-profiles ml-pp0-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access route $junos-framed-route-ip-address-prefix]
   user@host# **set next-hop $junos-framed-route-nexthop**
   user@host# **set metric $junos-framed-route-cost**
   user@host# **set preference $junos-framed-route-distance**

6. Configure the internal access route for the routing options.

   [edit dynamic-profiles ml-pp0-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access-internal route $junos-subscriber-ip-address**

7. Configure the qualified next-hop for the internal route.

   [edit dynamic-profiles ml-pp0-member-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access-internal route $junos-subscriber-ip-address]
   user@host# **set qualified-next-hop $junos-interface-name**

8. Configure the basic settings for the dynamic profile. See "Configuring a Dynamic Profile for Dynamic pp0 Member Link IFL Without Mixed Mode Support" on page 91, steps 5 through 11.

   NOTE: To enable mixed mode support, when the CPE is a PPP subscriber, you must also add an unnumbered address, and input and output filters to the **family inet** statement.

   [edit dynamic-profiles ml-pp0-member-prof interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
   user@host# **set family inet unnumbered-address $junos-loopback-interface**
   user@host# **set family inet filter input "$junos-input-filter" output "$junos-output-filter"**

9. When the CPE is a PPP subscriber, you must also configure class of service and define the traffic control profile.

[edit dynamic-profiles ml-pp0-member-prof class-of-service
user@host# **set traffic-control-profiles tc-profile**

10. For the traffic-control profile, define the following settings: scheduler map, shaping rate, overhead accounting, guaranteed rate, and delay buffer rate.

    [edit dynamic-profiles ml-pp0-member-prof 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 "$junos-cos-shaping-mode" bytes "$junos-cos-byte-adjust"**
    user@host# **set guaranteed-rate "$junos-cos-guaranteed-rate"**
    user@host# **set delay-buffer-rate "$junos-cos-delay-buffer-rate"**

11. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*.

    [edit dynamic-profiles ml-pp0-member-prof class-of-service]
    user@host# **set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”**

12. For the dynamic profile interface, define the following settings: output traffic control profile, classifiers, and rewrite rules.

    [edit dynamic-profiles ml-pp0-member-prof class-of-service interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
    user@host# **set output-traffic-control-profile tc-profile**
    user@host# **set classifiers dscp GEN-CLASSIFIER-IN**
    user@host# **set rewrite-rules dscp GEN-RW-OUT-DSCP**

13. If you are done configuring the device, commit the configuration.

    ```
    [edit]
    user@host# commit
    ```

Configuring a Dynamic Profile for the Dynamic Bundle IFL

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy.

To configure the dynamic profile for the dynamic bundle IFL, you specify the **encapsulation multilink-ppp** statement within the dynamic profile. The **dynamic profile** for the dynamic bundle IFL is referenced from the **dynamic profile** for dynamic PPPoE and LNS member link IFLs.

You must configure the **fragmentation-maps** statement statically using **class-of-service** and assign them in the bundle dynamic profile. You can also set these optional MLPPP parameters: MRRU, short sequence, and fragment-threshold. The following example shows how to configure the dynamic profile for the dynamic bundle IFL:

1. Specify the dynamic profile name for the bundle.

   [edit dynamic-profiles}
   user@host# **set ml-bundle-prof**

2. Although MLPPP member links process authentication and routing-instance assignments, if a non-default routing-instance is assigned, you must configure the bundle IFL under the assigned routing-instance. As a result, you must also configure routing-instances in the bundle dynamic-profile.

   [edit dynamic-profiles ml-bundle-prof]
   user@host# **set routing-instances "$junos-routing-instance" interface "$junos-interface-name"**

3. Configure the access route for the routing options.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access route $junos-framed-route-ip-address-prefix**

4. Configure the next-hop, metric, and preference for the router.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access route $junos-framed-route-ip-address-prefix]
   user@host# **set next-hop $junos-framed-route-nexthop**
   user@host# **set metric $junos-framed-route-cost**
   user@host# **set preference $junos-framed-route-distance**

5. Configure the internal access route for the routing options.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name"]
   user@host# **set routing-options access-internal route $junos-subscriber-ip-address**

6. Configure the qualified next-hop for the internal route.

   [edit dynamic-profiles ml-bundle-prof routing-instances "$junos-routing-instance" interface "$junos-interface-name" routing-options access-internal route $junos-subscriber-ip-address]

user@host# **set qualified-next-hop $junos-interface-name**

7. Configure the interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

   [edit dynamic-profiles ml-bundle-prof]
   user@host# **set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”**

8. Configure the **encapsulation multilink-ppp** statement to enable MLPPP bundling for the dynamic profile.

   [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
   user@host# **set encapsulation multilink-ppp**

9. Configure the following MLPPP options for this example:

   - **mrru**—Specifies the maximum received reconstructed unit value ranging from 1500 through 4500 bytes.
   - **fragment-threshold**—Applies to all packets and forwarding classes, ranging from 128 through 16,320 bytes.
   - **short-sequence**—Determines the header format for the MLPPP. Default is **long-sequence**.

   [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
   user@host# **set mrru 1500**
   user@host# **set fragment-threshold 320**
   user@host# **set short-sequence**

10. Enable support for MLPP subscribers.

    [edit dynamic-profiles ml-bundle-prof interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit”]
    user@host# **set family inet**

11. To enable **fragmentation-maps** support, you must configure class of service and define the traffic control profile.

    [edit dynamic-profiles ml-bundle-prof class-of-service]
    user@host# **set traffic-control-profiles tcp2**

12. For the traffic-control profile, define the following settings: scheduler map, shaping rate, guaranteed rate, and delay buffer rate.

    [edit dynamic-profiles ml-bundle-prof class-of-service traffic-control-profiles tcp2]
    user@host# **set scheduler-map "$junos-cos-scheduler-map"**
    user@host# **set shaping-rate "$junos-cos-shaping-rate"**
    user@host# **set guaranteed-rate "$junos-cos-guaranteed-rate"**
    user@host# **set delay-buffer-rate "$junos-cos-delay-buffer-rate"**

13. Configure the underlying interface for the dynamic profile by setting the predefined dynamic interface variable *$junos-interface-ifd-name*, and the underlying logical interface unit by setting the predefined unit number variable *$junos-interface-unit*. The interface and unit number variables are dynamically replaced with the interface and unit number that the subscriber accesses when connecting to the MX Series.

    [edit dynamic-profiles ml-bundle-prof class-of-service]
    user@host# **set interfaces “$junos-interface-ifd-name” unit "$junos-interface-unit"**

14. For the dynamic profile interface, define the output traffic control profile.

    [edit dynamic-profiles ml-bundle-prof class-of-service interfaces “$junos-interface-ifd-name” unit "$junos-interface-unit"]
    user@host# **set output-traffic-control-profile tcp2**

15. Define the fragmentation-map required for dynamic profile bundles and used to enable link fragmentation and interleaving (LFI).

    [edit dynamic-profiles ml-bundle-prof class-of-service interfaces “$junos-interface-ifd-name” unit "$junos-interface-unit"]
    user@host# **set fragmentation-map fragmap-2**

16. If you are done configuring the device, commit the configuration.

    [edit]
    user@host# **commit**

Results

From configuration mode, confirm your configuration by entering the **show dynamic-profiles** command with the sub-hierarchy levels **interfaces**. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces ge-1/0/0
  interfaces {
    ge-1/0/0 {
      flexible- vlan-tagging;
      unit 600 {
        encapsulation ppp-over-ether;
        vlan-id 600;
        pppoe-underlying-options {
          dynamic-profile ml-pp0-member-prot;
        }
      }
    }
  }
```

Dynamic profile for dynamic PPPoE member link IFL without mixed mode:

```
user@host# show dynamic-profiles mlp-pp0-member-profile
  dynamic-profile mlp-pp0-member-profile {
    interface $"$junos-interface-ifd-name" {
      unit "$junos-interface-unit" {
        pppoe-options {
          underlying-interface "$junos-underlying-interface";
```

```
            server;
          }
          ppp-options {
            pap;
            chap;
            lcp-restart-timer 5000;
          }
          family mlppp {
            bundle $junos-bundle-interface-name;
            service-interface si-5/1/0;
            dynamic-profile ml-bundle-prof;
          }
          family inet
        }
      }
    }
```

Dynamic profile for dynamic PPPoE member link IFL with mixed mode:

```
user@host# show dynamic-profiles mlp-pp0-member-profile
  dynamic-profile ml-pp0-member-prof {
    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 "$junos-interface-unit" {
        pppoe-options {
          underlying-interface "$junos-underlying-interface";
          server;
        }
        ppp-options {
          pap;
          chap;
          lcp-restart-timer 5000;
        }
        family mlppp {
          bundle $junos-bundle-interface-name;
          service-interface si-5/1/0;
          dynamic-profile ml-bundle-prof;
```

```
          }
          family inet {
            unnumbered-address $junos-loopback-interface;
            filter {
              input "$junos-input-filter";
              output "$junos-output-filter";
            }
          }
        }
    class-of-service {
      traffic-control-profiles {
          tc-profile {
            scheduler-map "$junos-cos-scheduler-map";
            shaping-rate "$junos-cos-shaping-rate";
          overhead-accounting "$junos-cos-shaping-mode" bytes "$junos-cos-byte-adjust";

            guaranteed-rate "$junos-cos-guaranteed-rate";
            delay-buffer-rate "$junos-cos-delay-buffer-rate";
          }
        }
        interfaces {
          "$junos-interface-ifd-name" {
            unit "$junos-interface-unit" {
              output-traffic-control-profile tc-profile;
              classifiers {
                dscp GEN-CLASSIFIER-IN;
              }
              rewrite-rules {
                dscp GEN-RW-OUT-DSCP;
              }
            }
          }
        }
      }
    }
```

Verification

Confirm that the configuration is working properly.

- Verifying the Subscriber Information on page 100
- Verifying Mixed Mode Support with a Dynamic MLPPP-Capable Subscriber on page 101
- Verifying Tunneled PPPoE MLPPP Interfaces on page 102

Verifying the Subscriber Information

Purpose Verify that the subscriber information for dynamic MLPPP over PPPoE is correct.

Action

```
root@haverhill> show subscribers extensive
Type: PPPoE
User Name: dual-stack-v4v6-nas@almach.com
Logical System: default
Routing Instance: default
Interface: pp0.1073741824
Interface type: Dynamic
Underlying Interface: ge-1/1/0.3000
Dynamic Profile Name: DS-lac-mlppp-link-ipv6
MAC Address: 30:00:64:30:00:02
State: Active
PPP State: Tunneled
Local IP Address: 210.1.1.1
Remote IP Address: 210.1.1.2
Radius Accounting ID: 5
Session ID: 5
Bundle Session ID: 6
VLAN Id: 3000
Login Time: 2013-03-28 15:42:30 PDT

Type: MLPPP
Logical System: default
Routing Instance: default
Interface: si-1/1/0.1073741825
Interface type: Dynamic
Underlying Interface: si-1/1/0.1073741825
Dynamic Profile Name: DS-mlppp-bundle-ipv6
State: Active
PPP State: Tunneled
Local IP Address: N/A
Remote IP Address: N/A
Radius Accounting ID: 6
Session ID: 6
Underlying Session ID: 5
Login Time: 2013-03-28 15:42:30 PDT
```

Meaning When a PPPoE MLPPP session is tunneled, the bundle and member link binding is maintained. The **PPP State** setting for both bundle and member link is set to **Tunneled**. Although there is no NCP negotiation over the bundle, the bundle session remains active.

Verifying Mixed Mode Support with a Dynamic MLPPP-Capable Subscriber

Purpose Verify that mixed-mode interfaces negotiated correctly for the single link PPP using a dynamic MLPPP-capable subscriber.

Action

```
root@haverhill> show interfaces extensive pp0.1073741832
Logical interface pp0.1073741832 (Index 489) (SNMP ifIndex 712)
   (Generation 299)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 40,
      Session AC name: haverhill1, Remote MAC address: 00:00:64:3c:01:02,
      Underlying interface: ge-1/0/0.44 (Index 376)
    Traffic statistics:
     Input  bytes  :                 1213
     Output bytes  :                 1672
     Input  packets:                   41
     Output packets:                   49
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
      Input  packets:                   0
      Output packets:                   0
    Local statistics:
     Input  bytes  :                  159
     Output bytes  :                 1424
     Input  packets:                   10
     Output packets:                   18
    Transit statistics:
     Input  bytes  :                 1054                    0 bps
     Output bytes  :                  248                    0 bps
     Input  packets:                   31                    0 pps
     Output packets:                   31                    0 pps
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
      Input  packets:                   0
      Output packets:                   0
  Keepalive settings: Interval 45 seconds, Up-count 1, Down-count 3
  LCP state: Opened
  NCP state: inet: Opened, inet6: Opened, iso: Not-configured, mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Success
    Protocol inet, MTU: 65531, Generation: 384, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Primary
        Destination: Unspecified, Local: 101.0.0.1, Broadcast: Unspecified,
        Generation: 297
    Protocol inet6, MTU: 65531, Generation: 385, Route table: 0
      Addresses, Flags: Is-Primary
        Destination: Unspecified, Local: 2030::1
    Generation: 298
        Destination: Unspecified, Local: fe80::2a0:a50f:fc64:6ef2
    Generation: 299
```

Meaning When a dynamic MLPPP-capable subscriber negotiates a single link PPP, the results are the same as a non-MLPPP subscriber; no bundle IFL or SDB session is created.

Verifying Tunneled PPPoE MLPPP Interfaces

Purpose Verify that the PPPoE MLPPP member link IFL is correct.

Action

```
root@haverhill> show interfaces extensive pp0.1073756921
Logical interface pp0.1073756921 (Index 482) (SNMP ifIndex 706)
   (Generation 15542)
    Flags: Up Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 37,
      Session AC name: haverhill, Remote MAC address: 00:00:64:1d:01:02,
      Underlying interface: ge-1/0/0.2040 (Index 457)
    Traffic statistics:
     Input  bytes  :                 273
     Output bytes  :                 270
     Input  packets:                  13
     Output packets:                  10
    Local statistics:
     Input  bytes  :                 138
     Output bytes  :                 155
     Input  packets:                   6
     Output packets:                   3
    Transit statistics:
     Input  bytes  :                 135                    0 bps
     Output bytes  :                 115                    0 bps
     Input  packets:                   7                    0 pps
     Output packets:                   7                    0 pps
  Keepalive settings: Interval 45 seconds, Up-count 1, Down-count 3
  LCP state: Opened
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
mpls:
  Not-configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 1492, Generation: 15534, Route table: 0
      Flags: Sendbcast-pkt-to-re
    Protocol mlppp, Multilink bundle: si-1/0/0.1073756922
    Service device pool: sipool-1, Dynamic profile: ml-bundle-prof
    MTU: 1526, Generation: 15533, Route table: 0
```

Meaning

When a PPPoE MLPPP session is tunneled, the bundle and member link binding remains. Although the bundle IFL does not participate in the control and forwarding path, it remains in the user interface.

Related Documentation

- MLPPP Support for LNS and PPPoE Subscribers on MX Series Overview on page 5
- Mixed Mode Support for MLPPP and PPP Subscribers Overview on page 9
- MLPPP Bundles and Inline Service Logical Interfaces Overview on page 11

CHAPTER 11

Dynamic PPP Subscriber Services Configuration

Configuring PPP Subscriber Services for MLPPP Bundles

You can configure PPP subscriber services for static LSQ MLPPP bundle interfaces.

To configure PPP subscriber services for static LSQ MLPPP bundle interfaces:

1. Enable PPP subscriber services for the interfaces.

   See "Enabling PPP Subscriber Services for Static Non-Ethernet Interfaces" on page 105.

2. Attach a dynamic profile to the MLPPP bundle interface.

   See "Attaching Dynamic Profiles to MLPPP Bundles" on page 106.

Related Documentation

- Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30
- Example: Minimum MLPPP Dynamic Profile on page 106
- Example: Configuring CoS on Static LSQ MLPPP Bundle Interfaces on page 107

Enabling PPP Subscriber Services for Static Non-Ethernet Interfaces

You can enable PPP subscriber services for certain non-Ethernet interface types on particular associated PICs. Supported interfaces are listed in "Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces" on page 30.

To enable PPP subscriber services on supported non-Ethernet interfaces:

- Configure PPP subscriber services.

```
[edit chassis]
user@host# set ppp-subscriber-services enable
```

To disable PPP subscriber services on supported non-Ethernet interfaces:

- Disable PPP subscriber services.

```
[edit chassis]
user@host# set ppp-subscriber-services disable
```

Related Documentation

- For hardware requirements, see Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30
- Configuring PPP Subscriber Services for MLPPP Bundles on page 105

Attaching Dynamic Profiles to MLPPP Bundles

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

To attach a dynamic profile to a static LSQ MLPPP bundle interface:

1. Specify that you want to configure PPP options.

```
[edit interfaces lsq-3/3/0 unit 0]
user@host# edit ppp-options
```

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

```
[edit interfaces lsq-3/3/0 unit 0 ppp-options]
user@host# set dynamic-profile vod-profile-50
```

Related Documentation

- Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30
- Configuring PPP Subscriber Services for MLPPP Bundles on page 105
- *Dynamic Profiles for PPP Subscriber Interfaces Overview*
- *Dynamic Profiles Overview*
- *Configuring a Basic Dynamic Profile*
- Configuring PPP Subscriber Services for MLPPP Bundles on page 105
- Example: Minimum MLPPP Dynamic Profile on page 106
- Example: Configuring CoS on Static LSQ MLPPP Bundle Interfaces on page 107

Example: Minimum MLPPP Dynamic Profile

This example shows the minimum configuration for a dynamic profile that is used for static LSQ MLPPP bundle interfaces.


```
dynamic-profiles {
  mlppp-profile-1 {
    interfaces {
      "$junos-interface-ifd-name" {
        unit "$junos-underlying-interface-unit";
      }
    }
  }
}
```

Related Documentation

- *Dynamic Profiles for PPP Subscriber Interfaces Overview*
- Attaching Dynamic Profiles to MLPPP Bundles on page 106

Example: Configuring CoS on Static LSQ MLPPP Bundle Interfaces

This example shows how to configure dynamic subscriber services on MLPPP bundle interfaces. The MLPPP bundles must be configured on link services intelligent queuing (IQ) (LSQ) interfaces. The MLPPP interfaces must be statically configured.

To configure dynamic subscriber services on static LSQ MLPPP bundle interfaces:

1. Configure class of service features for the LSQ interfaces.

```
[edit]
class-of-service
  classifiers {
    inet-precedence inet_classifier {
      forwarding-class best-effort {
        loss-priority low code-points 000;
      }
      forwarding-class expedited-forwarding {
        loss-priority low code-points 011;
      }
      forwarding-class assured-forwarding {
        loss-priority low code-points 100;
      }
    }
  }
  fragmentation-maps {
    sample-fragmap {
      forwarding-class {
        best-effort {
          fragment-threshold 1000;
          multilink-class 1:
        }
        assured-forwarding {
          fragment-threshold 1000;
          multilink-class 2:
        }
        expedited-forwarding {
          multilink-class 3:
        }
      }
    }
```

```
    }
    forwarding-classes {
        queue 0 best-effort;
        queue 1 expedited-forwarding;
        queue 2 assured-forwarding;
    }
    # traffic classifiers are statically defined
        network traffic interface{
            classifiers {
                inet-precedence inet_classifier;
            }
        }
    scheduler-maps {
        allthree {
            forwarding-class best-effort scheduler be-scheduler;
            forwarding-class expedited-forwarding scheduler hiprior-sched;
            forwarding-class assured-forwarding scheduler vpn-sched;
        }
    }
    schedulers {
        be-scheduler {
            transmit-rate percent 30;
            priority low;
        }
        hiprior-scheduler {
            transmit-rate percent 40;
            priority strict-high;
        }
        vpn-sched {
            transmit-rate percent 30;
            medium-high;
        }
    }
}
```

2. Configure the MLPPP bundle interfaces and the LSQ interfaces.

```
[edit interfaces]
t1-3/1/0:1:1 {
    keepalives interval 600;
    encapsulation ppp;
    unit 0 {
        ppp-options {
            lcp-restart-timer 5000;
        }
        family mlppp {
            bundle lsq-3/3/0.0;
        }
    }
}
t1-3/1/0:1:2 {
    keepalives interval 600;
    encapsulation ppp;
    unit 0 {
        ppp-options {
            lcp-restart-timer 5000;
```



```
            }
            family mlppp {
                bundle lsq-3/3/0.0;
            }
        }
    }
    lsq-3/3/0 {
        unit 0 {
            encapsulation multilink-ppp;
            multilink-max-classes 4;
            ppp-options {
                ncp-restart-timer 10000;
                dynamic-profile mlppp-profile;
            }
            family inet {
                address 192.168.1.1/32 {
                    destination 192.168.25.45;
                }
            }
        }
    }
```

3. Configure the dynamic profile that is applied to the MLPPP bundle interfaces.

```
[edit]
dynamic-profiles {
    mlppp-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit junos-underlying-interface-unit" {
                    family inet {
                        filter {
                            input "$junos-input-filter";
                            output "$junos-output-filter";
                        }
                    }
                }
            }
        }
        class-of-service {
            interfaces {
                "$junos-interface-ifd-name" {
                    unit junos-underlying-interface-unit" {
                        output-traffic-control-profile tcp1;
                        fragmentation-map sample-fragmap;
                    }
                }
            }
            traffic-control-profiles {
                tcp1 {
                    scheduler-map "junos-cos-scheduler-map";
                    shaping-rate "$junos-cos-shaping-rate";
                    guaranteed-rate "$junos-cos-guaranteed-rate";
                    delay-buffer-rate "$junos-cos-delay-buffer-rate";
                }
            }
```



```
            scheduler-maps {
                data_smap {
                    forwarding-class be scheduler data_sch;
                }
            }
            schedulers {
                be_sch {
                    ...
                }
            }
        }
    }
}
```

Related Documentation

- Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30
- *Layer 2 Service Package Capabilities and Interfaces*

CHAPTER 12

Configuration Statements

dynamic-profiles

Syntax

```
dynamic-profiles {
  profile-name {
    class-of-service {
      interfaces {
        interface-name ;
        }
        unit logical-unit-number {
          classifiers {
            type (classifier-name | default);
          }
          output-traffic-control-profile (profile-name | $junos-cos-traffic-control-profile);
          rewrite-rules {
            dscp (rewrite-name | default);
            dscp-ipv6 (rewrite-name | default);
            ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner);
            inet-precedence (rewrite-name | default);
            }
          }
        }
      }
      scheduler-maps {
        map-name {
          forwarding-class class-name scheduler scheduler-name;
        }
      }
      schedulers {
        (scheduler-name) {
          buffer-size (seconds | percent percentage | remainder | temporal microseconds);
          drop-profile-map loss-priority (any | low | medium-low | medium-high | high)
            protocol (any | non-tcp | tcp) drop-profile profile-name;
          excess-priority (low | high | $junos-cos-scheduler-excess-priority);
          excess-rate (percent percentage | percent $junos-cos-scheduler-excess-rate);
          overhead-accounting (shaping-mode) <bytes (byte-value>;
          priority priority-level;
          shaping-rate (rate | predefined-variable);
          transmit-rate (percent percentage | rate | remainder) <exact | rate-limit>;
        }
      }
      traffic-control-profiles profile-name {
        delay-buffer-rate (percent percentage | rate | $junos-cos-delay-buffer-rate);
        excess-rate (percent percentage | proportion value | percent $junos-cos-excess-rate);
        guaranteed-rate (percent percentage | rate | $junos-cos-guaranteed-rate);
        overhead-accounting (shaping-mode) <bytes (byte-value>;
        scheduler-map map-name;
        shaping-rate (rate | predefined-variable);
      }
    }
    firewall {
      family family {
        fast-update-filter filter-name {
          interface-specific;
          match-order [match-order];
```



```
        term term-name {
            from {
                match-conditions;
            }
            then {
                action;
                action-modifiers;
            }
            only-at-create;
        }
    }
    firewall {
        family family {
            fast-update-filter filter-name {
                interface-specific;
            match-order [match-order];
                term term-name {
                    from {
                        match-conditions;
                    }
                    then {
                        action;
                        action-modifiers;
                    }
                only-at-create;
            filter filter-name {
            interface-specific;
                term term-name {
                    from {
                        match-conditions;
                    }
                    then {
                        action;
                        action-modifiers;
                    }
                }
        policer policer-name {
            filter-specific;
            if-exceeding {
                (bandwidth-limit bps | bandwidth-percent percentage);
                burst-size-limit bytes;
            }
            logical-bandwidth-policer;
            logical-interface-policer;
            physical-interface-policer;
            then {
                policer-action;
            }
        }
        hierarchical-policer policer-name {
            aggregate {
                if-exceeding {
                    bandwidth-limit-limit bps;
                    burst-size-limit bytes;
                }
                then {
```



```
                policer-action;
              }
            }
            premium {
              if-exceeding {
                bandwidth-limit bps;
                burst-size-limit bytes;
              }
              then {
                policer-action;
              }
            }
          }
          three-color-policer policer-name {
            action {
              loss-priority high then discard;
            }
            logical-interface-policer;
            single-rate {
              (color-aware | color-blind);
              committed-burst-size bytes;
              committed-information-rate bps;
              excess-burst-size bytes;
            }
            two-rate {
              (color-aware | color-blind);
              committed-burst-size bytes;
              committed-information-rate bps;
              peak-burst-size bytes;
              peak-information-rate bps;
              }
            }
          }
        }
        policy-options {
          prefix-listname {
            ip-addresses;
          }
        }
      }
    }
    interfaces interface-name {
      interface-set interface-set-name {
        interface interface-name {
          unit logical unit number {
            advisory-options {
              downstream-rate rate;
              upstream-rate rate;
            }
          }
        }
      }
      unit logical-unit-number {
        auto-configure {
          agent-circuit-identifier {
            dynamic-profile profile-name;
```



```
  }
}
encapsulation (atm-ccc-cell-relay | atm-ccc-vc-mux | atm-cisco-nlpid |
  atm-tcc-vc-mux | atm-mlppp-llc | atm-nlpid | atm-ppp-llc | atm-ppp-vc-mux |
  atm-snap | atm-tcc-snap | atm-vc-mux | ether-over-atm-llc |
  ether-vpls-over-atm-llc | ether-vpls-over-fr | ether-vpls-over-ppp | ethernet |
  frame-relay-ccc | frame-relay-ppp | frame-relay-tcc | frame-relay-ether-type |
  frame-relay-ether-type-tcc | multilink-frame-relay-end-to-end | multilink-ppp |
  ppp-over-ether | ppp-over-ether-over-atm-llc | vlan-bridge | vlan-ccc | vlan-vci-ccc
  | vlan-tcc | vlan-vpls);
family family {
  address address;
  filter {
    adf {
      counter;
      input-precedence precedence;
      not-mandatory;
      output-precedence precedence;
      rule rule-value;
    }
    input filter-name (
      precedence precedence;
    }
    output filter-name {
      precedence precedence;
    }
  }
  rpf-check {
    fail-filter filter-name;
    mode loose;
  }
  service {
    input {
      service-set service-set-name {
        service-filter filter-name;
      }
      post-service-filter filter-name;
    }
    input-vlan-map {
      inner-tag-protocol-id tpid;
      inner-vlan-id number;
      (push | swap);
      tag-protocol-id tpid;
      vlan-id number;
    }
    output {
      service-set service-set-name {
        service-filter filter-name;
      }
    }
    output-vlan-map {
      inner-tag-protocol-id tpid;
      inner-vlan-id number;
      (pop | swap);
      tag-protocol-id tpid;
      vlan-id number;
```

```
            }
          }
          unnumbered-address interface-name <preferred-source-address address>;
        }
        ppp-options {
          chap;
          pap;
        }
        vlan-id number;
        vlan-tags outer [tpid].vlan-id [inner [tpid].vlan-id];
      }
    }
    interfaces {
      demux0 {...}
    }
    interfaces {
      pp0 {...}
    }
    protocols {
      igmp {
        interface interface-name {
          accounting;
          disable;
          group-policy;
          immediate-leave
          no-accounting;
          promiscuous-mode;
          ssm-map ssm-map-name;
          static {
            group group {
              source source;
            }
          }
          version version;
        }
        mld {
          interfaceinterface-name {
            disable;
            (accounting | no-accounting);
            group-policy;
            immediate-leave;
            oif-map;
            passive;
            ssm-map ssm-map-name;
            static {
              group multicast-group-address {
                exclude;
                group-count number;
                group-increment increment;
                source ip-address {
                  source-count number;
                  source-increment increment;
                }
              }
            }
            version version;
```



```
            }
        }
        router-advertisement {
            interface interface-name {
                current-hop-limit number;
                default-lifetime seconds;
                (managed-configuration | no-managed-configuration);
                max-advertisement-interval seconds;
                min-advertisement-interval seconds;
                (other-stateful-configuration | no-other-stateful-configuration);
                prefix prefix;
                reachable-time milliseconds;
                retransmit-timer milliseconds;
            }
        }
    }
}
routing-instances routing-instance-name {
    interface interface-name;
    routing-options {
        access {
            route prefix {
                next-hop next-hop;
                metric route-cost;
                preference route-distance;
                tag route-tag;
            }
        }
        access-internal {
            route subscriber-ip-address {
                qualified-next-hop underlying-interface {
                    mac-address address;
                }
            }
        }
        multicast {
            interface interface-name {
                no-qos-adjust;
            }
        }
    }
    rib routing-table-name {
        access {
            route prefix {
                next-hop next-hop;
                metric route-cost;
                preference route-distance;
                tag route-tag;
            }
        }
        access-internal {
            route subscriber-ip-address {
                qualified-next-hop underlying-interface {
                    mac-address address;
                }
            }
```

```
            }
          }
        }
        routing-options {
          access {
            route prefix {
              next-hop next-hop;
              metric route-cost;
              preference route-distance;
              tag route-tag;
            }
          }
          access-internal {
            route subscriber-ip-address {
              qualified-next-hop underlying-interface {
                mac-address address;
              }
            }
          }
          multicast {
            interface interface-name {
              no-qos-adjust;
            }
          }
        }
        variables {
          variable-name {
            default-value default-value;
            equals expression;
            mandatory;
            uid;
            uid-reference;
          }
        }
      }
    }
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 9.2.
Support at the **filter, policer, hierarchical-policer, three-color-policer, and policy options** hierarchy levels introduced in Junos OS Release 11.4.

Description Create dynamic profiles for use with DHCP or PPP client access.

Options ***profile-name***—Name of the dynamic profile; string of up to 80 alphanumeric characters.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring a Basic Dynamic Profile*
- *Configuring Dynamic VLANs Based on Agent Circuit Identifier Information*
- *Dynamic Profiles Overview*

forwarding-classes (Class-of-Service)

Syntax

```
forwarding-classes {
  class queue-num queue-number priority (high | low);
  queue queue-number class-name priority (high | low) [ policing-priority (premium | normal) ];
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.
policing-priority option introduced in Junos OS Release 9.5.
Statement introduced on PTX Series Packet Transport Routers in Junos OS Release 12.1.

Description Associate the forwarding class with a queue name and number. For M320, MX Series, T Series routers and EX Series switches only, you can configure fabric priority queuing by including the **priority** statement. For Enhanced IQ PICs, you can include the **policing-priority** option.

NOTE: **The priority add policing-priority options are not supported on PTX Series Packet Transport Routers.**

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Forwarding Classes*
- *Forwarding Classes and Fabric Priority Queues*
- *Configuring Layer 2 Policers on IQE PICs*
- *Classifying Packets by Egress Interface*

fragmentation-maps

Syntax

```
fragmentation-maps {
  map-name {
    forwarding-class class-name {
      (fragment-threshold bytes | no-fragmentation);
      multilink-class number;
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.

Description For link services IQ (**lsq**) interfaces only, define fragmentation properties for individual forwarding classes.

Default If you do not include this statement, traffic in all forwarding classes is fragmented.

Options ***map-name***—Name of the fragmentation map.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces*

inline-services (PIC level)

Syntax

```
inline-services {
  bandwidth (1g | 10g);
}
```

Hierarchy Level [edit chassis fpc *slot-number* pic *number*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Enable inline services on PICs residing on MPCs. To enable inline services that are specified at the fpc level, see configuration statement *inline-services (FPC Level)*

The remaining statement is explained separately.

Options The option is described separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Enabling Inline Service Interfaces*
- *Configuring an L2TP LNS with Inline Service Interfaces*

interfaces (Class of Service)

Syntax

```
interfaces {
  mif. number {
    unit logical-unit-number {
      ingress-rewrite-rules {
        [dscp (rewrite-rule-name | default)];
        [dscp-ipv6 (rewrite-rule-name | default)];
        [inet-precedence (rewrite-rule-name | default)];
        }
      }
      rewrite-rules {
        [dscp (rewrite-rule-name | default)] {
          protocol [(gtp-inet-both | gtp-inet-outer)];
        }
        [dscp-ipv6 (rewrite-rule-name | default)] {
          protocol [(mpls | gtp-inet-both | gtp-inet-outer)];
        }
        [inet-precedence (rewrite-rule-name | default)] {
          protocol [(gtp-inet-both | gtp-inet-outer)];
        }
      }
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Mobility Release 11.2W.

Description Specify the mobile interfaces to set the CoS values for upstream and downstream subscriber packets.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Applying Rewrite Rules on Mobile Interfaces Overview*
- *Applying Ingress Rewrite Rules to a Mobile Interface*
- *class-of-service (MobileNext Broadband Gateway)*

ppp-subscriber-services

Syntax ppp-subscriber-services (disable | enable);

Hierarchy Level [edit chassis]

Release Information Statement introduced in Junos OS Release 10.2.

Description Enable dynamic PPP subscriber services on non-PPPoE interfaces on certain PICs.

NOTE: When you include this statement, the relevant PICs restart. This action disrupts subscribers already logged in through those PICs. You can confirm completion of the restart by issuing the **show chassis pic fpc-slot *slot-number* pic-slot *slot-number*** command.

Options **disable**—Disable subscriber services.

enable—Enable subscriber services.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *show chassis pic*
- Attaching Dynamic Profiles to MLPPP Bundles on page 106
- For hardware requirements, see Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces on page 30

profile (Access)

Syntax

```
profile profile-name {
  accounting {
    address-change-immediate-update
    accounting-stop-on-access-deny;
    accounting-stop-on-failure;
    ancp-speed-change-immediate-update;
    coa-immediate-update;
    coa-no-override service-class-attribute;
    duplication;
    duplication-vrf {
      access-profile-name profile-name;
      vrf-name vrf-name;
    }
    immediate-update;
    order [ accounting-method ];
    send-acct-status-on-config-change;
    statistics (time | volume-time);
    update-interval minutes;
    wait-for-acct-on-ack;
    }
  authentication-order [ authentication-methods ];
  client client-name {
    chap-secret chap-secret;
    group-profile profile-name;
    ike {
      allowed-proxy-pair {
        remote remote-proxy-address local local-proxy-address;
      }
      pre-shared-key (ascii-text character-string | hexadecimal hexadecimal-digits);
      ike-policy policy-name;
      interface-id string-value;
    }
    l2tp {
      aaa-access-profile profile-name;
      interface-id interface-id;
      lcp-renegotiation;
      local-chap;
      maximum-sessions-per-tunnel number;
      multilink {
        drop-timeout milliseconds;
        fragment-threshold bytes;
      }
      ppp-authentication (chap | pap);
      ppp-profile profile-name;
      shared-secret shared-secret;
    }
    pap-password pap-password;
    ppp {
      cell-overhead;
      encapsulation-overhead bytes;
      framed-ip-address ip-address;
      framed-pool framed-pool;
```

```
    idle-timeout seconds;
    interface-id interface-id;
    keepalive seconds;
    primary-dns primary-dns;
    primary-wins primary-wins;
    secondary-dns secondary-dns;
    secondary-wins secondary-wins;
  }
  user-group-profile profile-name;
}
domain-name-server;
domain-name-server-inet;
domain-name-server-inet6;
preauthentication-order preauthentication-method;
provisioning-order (gx-plus | jsrc);
radius {
  accounting-server [ ip-address ];
  attributes {
    exclude {
      ...
    }
    ignore {
      framed-ip-netmask;
      input-filter;
      logical-system:routing-instance;
      output-filter;
    }
  }
  authentication-server [ ip-address ];
  options {
    accounting-session-id-format (decimal | description);
    calling-station-id-delimiter delimiter-character;
    calling-station-id-format {
      agent-circuit-id;
      agent-remote-id;
      interface-description;
      nas-identifier;
    }
    client-accounting-algorithm (direct | round-robin);
    client-authentication-algorithm (direct | round-robin);
    coa-dynamic-variable-validation;
    ethernet-port-type-virtual;
    interface-description-format {
      exclude-adapter;
      exclude-sub-interface;
    }
    juniper-dsl-attributes;
    nas-identifier identifier-value;
    nas-port-extended-format {
      adapter-width width;
      ae-width width;
      port-width width;
      slot-width width;
      stacked-vlan-width width;
      vlan-width width;
      atm {
```



```
                adapter-width width;
                port-width width:
                slot-width width;
                vci-width width:
                vpi-width width;
            }
        }
        nas-port-id-delimiter delimiter-character;
        nas-port-id-format {
            agent-circuit-id;
            agent-remote-id;
            interface-description;
            nas-identifier;
        }
        nas-port-type {
            ethernet {
                port-type;
            }
        }
        revert-interval interval;
        vlan-nas-port-stacked-format;
    }
    preauthentication-server ip-address;
}
radius-server server-address {
    accounting-port port-number;
    accounting-retry number;
    accounting-timeout seconds;
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    secret password;
    max-outstanding-requests value;
    source-address source-address;
    timeout seconds;
}
service {
    accounting-order (activation-protocol | radius);
}
session-options {
    client-group [ group-names ];
    client-idle-timeout minutes;
    client-session-timeoutminutes;
}
}
```

Hierarchy Level [edit access]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure PPP CHAP, or a profile and its subscriber access, L2TP, or PPP properties.

Options

profile-name—Name of the profile.

For CHAP, the name serves as the mapping between peer identifiers and CHAP secret keys. This entity is queried for the secret key whenever a CHAP challenge or response is received.

The remaining statements are explained separately.

Required Privilege Level

admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- *Configuring the PPP Authentication Protocol*
- *Configuring Access Profiles for L2TP or PPP Parameters*
- *Configuring L2TP Properties for a Client-Specific Profile*
- *Configuring an L2TP LNS with Inline Service Interfaces*
- *Configuring PPP Properties for a Client-Specific Profile*
- *Configuring Service Accounting with JSRC*
- *AAA Service Framework Overview*
- *show network-access aaa statistics*
- *clear network-access aaa statistics*

schedulers (Class of Service)

Syntax

```
schedulers {
  scheduler-name {
    adjust-minimum rate;
    adjust-percent percentage;
    buffer-size (seconds | percent percentage | remainder | temporal microseconds);
    drop-profile-map loss-priority (any | low | medium-low | medium-high | high) protocol
      (any | non-tcp | tcp) drop-profile profile-name;
    excess-priority [ low | medium-low | medium-high | high | none];
    excess-rate (percent percentage | proportion value);
    priority priority-level;
    shaping-rate (percent percentage | rate);
    transmit-rate (percent percentage | rate | remainder) <exact | rate-limit>;
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1X48 for PTX Series routers.

Description Specify the scheduler name and parameter values.

Options ***scheduler-name***—Name of the scheduler to be configured.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Schedulers Overview*
- *Default Schedulers Overview*
- *Configuring Schedulers*
- *Configuring a Scheduler*

service-device-pool (L2TP)

Syntax service-device-pool *pool-name*;

Hierarchy Level [edit services l2tp tunnel-group *name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Assign a pool of service interfaces to the tunnel group to balance traffic across.

NOTE: The service interface configuration is required for static LNS sessions. Either the service interface configuration or the service device pool configuration can be used for dynamic LNS sessions.

Options *pool-name*—Name of the service device pool.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring an L2TP Tunnel Group for LNS Sessions with Inline Services Interfaces*

traffic-control-profiles

Syntax

```
traffic-control-profiles profile-name {
  adjust-minimum rate;
  atm-service (cbr | rtvbr | nrtvbr);
  delay-buffer-rate (percent percentage | rate);
  excess-rate (percent percentage  | proportion value );
  excess-rate-high (percent percentage | proportion value);
  excess-rate-low (percent percentage | proportion value);
  guaranteed-rate (percent percentage | rate) <burst-size bytes>;
  max-burst-size cells;
  overhead-accounting (frame-mode | cell-mode | frame-mode-bytes | cell-mode-bytes)
    <bytes (byte-value)>;
  peak-rate rate;
  scheduler-map map-name;
  shaping-rate (percent percentage | rate) <burst-size bytes>;
  shaping-rate-excess-high rate [ burst-size bytes ];
  shaping-rate-excess-low rate [ burst-size bytes ];
  shaping-rate-priority-high rate [ burst-size bytes ];
  shaping-rate-priority-low rate [ burst-size bytes ];
  shaping-rate-priority-medium rate [ burst-size bytes ];
  strict-priority-scheduler;
  sustained-rate rate;
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 7.6.

Description For Gigabit Ethernet IQ, Channelized IQ PICs, FRF.15 and FRF.16 LSQ interfaces, Enhanced Queuing (EQ) DPCs, and PTX Series routers only, configure traffic shaping and scheduling profiles. For Enhanced EQ PICs, EQ DPCs, and PTX Series routers only, you can include the **excess-rate** statement.

Options ***profile-name***—Name of the traffic-control profile.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Oversubscribing Interface Bandwidth*
- *Understanding Scheduling on PTX Series Routers*
- *output-traffic-control-profile*

PART 3

Administration

CHAPTER 13

Operational Commands

show interfaces (PPPoE)

Syntax

show interfaces pp0.*logical*
<brief | detail | extensive | terse>
<descriptions>
<media>
<snmp-index *snmp-index*>
<statistics>

Release Information Command introduced before Junos OS Release 7.4.

Description (J Series Services Routers, M120 routers, M320 routers, and MX Series routers only) Display status information about the PPPoE interface.

Options

pp0.*logical*—Display standard status information about the PPPoE interface.

brief | detail | extensive | terse—(Optional) Display the specified level of output.

descriptions—(Optional) Display interface description strings.

media—(Optional) Display media-specific information about PPPoE interfaces.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display PPPoE interface statistics.

Required Privilege Level view

List of Sample Output

show interfaces (PPPoE) on page 140
show interfaces (PPPoE over Aggregated Ethernet) on page 140
show interfaces brief (PPPoE) on page 141
show interfaces detail (PPPoE) on page 141
show interfaces detail (PPPoE on J Series Services Routers) on page 142
show interfaces extensive (PPPoE on M120 and M320 Routers) on page 143

Output Fields Table 3 on page 134 lists the output fields for the **show interfaces** (PPPoE) command. Output fields are listed in the approximate order in which they appear.

Table 3: show interfaces (PPPoE) Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels
Enabled	State of the interface. Possible values are described in the “Enabled Field” section under *Common Output Fields Description*.	All levels
Interface index	Physical interface index number, which reflects its initialization sequence.	**detail extensive** none

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Type	Physical interface type (PPPoE).	All levels
Link-level type	Encapsulation on the physical interface (PPPoE).	All levels
MTU	MTU size on the physical interface.	All levels
Clocking	Reference clock source. It can be **Internal** or **External**.	All levels
Speed	Speed at which the interface is running.	All levels
Device flags	Information about the physical device. Possible values are described in the “Device Flags” section under *Common Output Fields Description*.	All levels
Interface flags	Information about the interface. Possible values are described in the “Interface Flags” section under *Common Output Fields Description*.	All levels
Link type	Physical interface link type: **full duplex** or **half duplex**.	All levels
Link flags	Information about the interface. Possible values are described in the “Link Flags” section under *Common Output Fields Description*.	All levels
Input rate	Input rate in bits per second (bps) and packets per second (pps).	None specified
Output rate	Output rate in bps and pps.	None specified
Physical Info	Physical interface information.	All levels
Hold-times	Current interface hold-time up and hold-time down, in milliseconds.	**detail extensive**
Current address	Configured MAC address.	**detail extensive**
Hardware address	MAC address of the hardware.	**detail extensive**
Alternate link address	Backup address of the link.	**detail extensive**
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the physical interface if IPv6 statistics tracking is enabled. NOTE: These fields include dropped traffic and exception traffic, as those fields are not separately defined. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
Input errors	Input errors on the interface: • **Errors**—Sum of incoming frame aborts and FCS errors. • **Drops**—Number of packets dropped by the input queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Framing errors**—Number of packets received with an invalid frame checksum (FCS). • **Runts**—Number of frames received that are smaller than the runt threshold. • **Giants**—Number of frames received that are larger than the giant threshold. • **Policed discards**—Number of frames that the incoming packet match code discarded because they were not recognized or not of interest. Usually, this field reports protocols that the Junos OS does not handle. • **Resource errors**—Sum of B chip Tx drops and IXP Tx net transmit drops.	**extensive**
Output errors	Output errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Carrier transitions** —Number of times the interface has gone from **down** to **up**. This number does not normally increment quickly, increasing only when the cable is unplugged, the far-end system is powered down and then up, or another problem occurs. If the number of carrier transitions increments quickly (perhaps once every 10 seconds), then the cable, the far-end system, or the PIM is malfunctioning. • **Errors**—Sum of the outgoing frame aborts and FCS errors. • **Drops**—Number of packets dropped by the output queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of B chip Tx drops and IXP Tx net transmit drops.	**extensive**
Logical Interface		

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
Logical interface	Name of the logical interface.	All levels
Index	Logical interface index number (which reflects its initialization sequence).	**detail extensive** none
SNMP ifIndex	Logical interface SNMP interface index number.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Flags	Information about the logical interface. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	All levels
Encapsulation	Type of encapsulation configured on the logical interface.	All levels
PPP parameters	PPP status: • LCP restart timer—Length of time (in milliseconds) between successive Link Control Protocol (LCP) configuration requests. • NCP restart timer—Length of time (in milliseconds) between successive Network Control Protocol (NCP) configuration requests.	**detail**
PPPoE	PPPoE status: • **State**—State of the logical interface (**up** or **down**). • **Session ID**—PPPoE session ID. • **Service name**—Type of service required. Can be used to indicate an Internet service provider (ISP) name or a class or quality of service. • **Configured AC name**—Configured access concentrator name. • **Auto-reconnect timeout**—Time after which to try to reconnect after a PPPoE session is terminated, in seconds. • **Idle Timeout**—Length of time (in seconds) that a connection can be idle before disconnecting. • **Underlying interface**—Interface on which PPPoE is running.	All levels
Link	Name of the physical interfaces for member links in an aggregated Ethernet bundle for a PPPoE over aggregated Ethernet configuration. PPPoE traffic goes out on these interfaces.	All levels
Traffic statistics	Total number of bytes and packets received and transmitted on the logical interface. These statistics are the sum of the local and transit statistics. When a burst of traffic is received, the value in the output packet rate field might briefly exceed the peak cell rate. This counter usually takes less than 1 second to stabilize.	**detail extensive**

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled. NOTE: The packet and byte counts in these fields include traffic that is dropped and does not leave the router. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**detail extensive**
Local statistics	Statistics for traffic received from and transmitted to the Routing Engine. When a burst of traffic is received, the value in the output packet rate field might briefly exceed the peak cell rate. This counter usually takes less than 1 second to stabilize.	**detail extensive**
Transit statistics	Statistics for traffic transiting the router. When a burst of traffic is received, the value in the output packet rate field might briefly exceed the peak cell rate. This counter usually takes less than 1 second to stabilize. NOTE: The packet and byte counts in these fields include traffic that is dropped and does not leave the router.	**detail extensive**
Keepalive settings	(PPP and HDLC) Configured settings for keepalives. • **interval *seconds***—The time in seconds between successive keepalive requests. The range is **10** seconds through **32,767** seconds, with a default of **10** seconds. • **down-count*number***—The number of keepalive packets a destination must fail to receive before the network takes a link down. The range is **1** through **255**, with a default of **3**. • **up-count *number***—The number of keepalive packets a destination must receive to change a link's status from down to up. The range is **1** through **255**, with a default of **1**.	**detail extensive**

Keepalive statistics	(PPP and HDLC) Information about keepalive packets.
	
	• **Input**—Number of keepalive packets received by PPP.
	  ◦ **(last seen 00:00:00 ago)**—Time the last keepalive packet was received, in the format ***hh*:*mm*:*ss***.
	• **Output**—Number of keepalive packets sent by PPP and how long ago the last keepalive packets were sent and received.
	  ◦ **(last seen 00:00:00 ago)**—Time the last keepalive packet was sent, in the format ***hh*:*mm*:*ss***.
	
	(MX Series routers with MPCs/MICs) When an MX Series router with MPCs/MICs is using PPP fast keepalive for a PPP link, the display does not include the number of keepalive packets received or sent, or the amount of time since the router received or sent the last keepalive packet.
		detail extensive

Input packets	Number of packets received on the logical interface.	None specified
Output packets	Number of packets transmitted on the logical interface.	None specified

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
LCP state	(PPP) Link Control Protocol state. • **Conf-ack-received**—Acknowledgement was received. • **Conf-ack-sent**—Acknowledgement was sent. • **Conf-req-sent**—Request was sent. • **Down**—LCP negotiation is incomplete (not yet completed or has failed). • **Not-configured**—LCP is not configured on the interface. • **Opened**—LCP negotiation is successful.	none **detail extensive**
NCP state	(PPP) Network Control Protocol state. • **Conf-ack-received**—Acknowledgement was received. • **Conf-ack-sent**—Acknowledgement was sent. • **Conf-req-sent**—Request was sent. • **Down**—NCP negotiation is incomplete (not yet completed or has failed). • **Not-configured**—NCP is not configured on the interface. • **Opened**—NCP negotiation is successful.	**detail extensive** none
CHAP state	(PPP) Displays the state of the Challenge Handshake Authentication Protocol (CHAP) during its transaction. • **Chap-Chal-received**—Challenge was received but response not yet sent. • **Chap-Chal-sent**—Challenge was sent. • **Chap-Resp-received**—Response was received for the challenge sent, but CHAP has not yet moved into the Success state. (Most likely with RADIUS authentication.) • **Chap-Resp-sent**—Response was sent for the challenge received. • **Closed**—CHAP authentication is incomplete. • **Failure**—CHAP authentication failed. • **Not-configured**—CHAP is not configured on the interface. • **Success**—CHAP authentication was successful.	none **detail extensive**
Protocol	Protocol family configured on the logical interface.	**detail extensive** none
protocol-family	Protocol family configured on the logical interface. If the protocol is **inet**, the IP address of the interface is also displayed.	**brief**
MTU	MTU size on the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Route table	Routing table in which the logical interface address is located. For example, **0** refers to the routing table inet.0.	**detail extensive** none
Flags	Information about the protocol family flags. Possible values are described in the “Family Flags” section under *Common Output Fields Description*.	**detail extensive** none

Table 3: show interfaces (PPPoE) Output Fields *(continued)*

Field Name	Field Description	Level of Output
Addresses, Flags	Information about the addresses configured for the protocol family. Possible values are described in the “Addresses Flags” section under *Common Output Fields Description*.	**detail extensive** none
Destination	IP address of the remote side of the connection.	**detail extensive** none
Local	IP address of the logical interface.	**detail extensive** none
Broadcast	Broadcast address.	**detail extensive** none

Sample Output

show interfaces (PPPoE)

```
user@host> show interfaces pp0
Physical interface: pp0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 24
  Type: PPPoE, Link-level type: PPPoE, MTU: 1532
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)

  Logical interface pp0.0 (Index 72) (SNMP ifIndex 72)
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionDown, Session ID: None,
      Service name: None, Configured AC name: sapphire,
      Auto-reconnect timeout: 100 seconds, Idle timeout: Never,
      Underlying interface: at-5/0/0.0 (Index 70)
  Input packets : 0
  Output packets: 0
  LCP state: Not-configured
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
  mpls: Not-configured
  CHAP state: Closed
    Protocol inet, MTU: 100
      Flags: User-MTU, Negotiate-Address
```

show interfaces (PPPoE over Aggregated Ethernet)

```
user@host> show interfaces pp0.1073773821
Logical interface pp0.1073773821 (Index 80) (SNMP ifIndex 32584)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 1,
      Session AC name: alcor, Remote MAC address: 00:10:94:00:00:01,
      Underlying interface: demux0.100 (Index 88)
    Link:
      ge-1/0/0.32767
      ge-1/0/1.32767
    Input packets : 6
```



```
    Output packets: 6
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured,  mpls:
Not-configured
  CHAP state: Closed
  PAP state: Success
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Primary
        Local: 45.63.24.1
```

show interfaces brief (PPPoE)

```
user@host> show interfaces pp0 brief
Physical interface: pp0, Enabled, Physical link is Up
  Type: PPPoE, Link-level type: PPPoE, MTU: 1532, Speed: Unspecified
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps

  Logical interface pp0.0
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionDown, Session ID: None,
      Service name: None, Configured AC name: sapphire,
      Auto-reconnect timeout: 100 seconds, Idle timeout: Never,
      Underlying interface: at-5/0/0.0 (Index 70)
    inet
```

show interfaces detail (PPPoE)

```
user@host> show interfaces pp0 detail
Physical interface: pp0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 24, Generation: 9
  Type: PPPoE, Link-level type: PPPoE, MTU: 1532, Speed: Unspecified
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  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
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
  Logical interface pp0.0 (Index 72) (SNMP ifIndex 72) (Generation 14)
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionDown, Session ID: None,
      Service name: None, Configured AC name: sapphire,
      Auto-reconnect timeout: 100 seconds, Idle timeout: Never,
      Underlying interface: at-5/0/0.0 (Index 70)
    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                    0 bps
     Output bytes  :                0                    0 bps
     Input  packets:                0                    0 pps
     Output packets:                0                    0 pps
  LCP state: Not-configured
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
  mpls: Not-configured
  CHAP state: Closed
    Protocol inet, MTU: 100, Generation: 14, Route table: 0
      Flags: User-MTU, Negotiate-Address
```

show interfaces detail (PPPoE on J Series Services Routers)

```
user@host> show interfaces pp0 detail
Physical interface: pp0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 24, Generation: 9
  Type: PPPoE, Link-level type: PPPoE, MTU: 1532, Speed: Unspecified
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  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
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                0                    0 bps
   Output bytes  :                0                    0 bps
   Input  packets:                0                    0 pps
   Output packets:                0                    0 pps
  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 pp0.0 (Index 72) (SNMP ifIndex 72) (Generation 14)
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionDown, Session ID: None,
      Service name: None, Configured AC name: sapphire,
      Auto-reconnect timeout: 100 seconds, Idle timeout: Never,
      Underlying interface: at-5/0/0.0 (Index 70)
    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                    0 bps
      Output bytes  :                   0                    0 bps
      Input  packets:                   0                    0 pps
      Output packets:                   0                    0 pps
   LCP state: Not-configured
   NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
   mpls: Not-configured
   CHAP state: Closed
     Protocol inet, MTU: 100, Generation: 14, Route table: 0
       Flags: User-MTU, Negotiate-Address
```

show interfaces extensive (PPPoE on M120 and M320 Routers)

```
user@host> show interfaces pp0 extensive
Physical interface: pp0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 93, Generation: 129
  Type: PPPoE, Link-level type: PPPoE, MTU: 1532, Speed: Unspecified
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps
  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
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              972192                    0 bps
   Output bytes  :              975010                    0 bps
   Input  packets:                1338                    0 pps
   Output packets:                1473                    0 pps
   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 pp0.0 (Index 69) (SNMP ifIndex 96) (Generation 194)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 26,
      Session AC name: None, AC MAC address: 00:17:cb:48:c8:12,
      Service name: None, Configured AC name: None,
      Auto-reconnect timeout: Never, Idle timeout: Never,
      Underlying interface: ge-3/0/1.0 (Index 67)
    Traffic statistics:
     Input  bytes  :                252
     Output bytes  :                296
     Input  packets:                  7
     Output packets:                  8
     IPv6 transit statistics:
      Input  bytes  :                 0
      Output bytes  :                 0
      Input  packets:                 0
```

```
      Output packets:                  0
    Local statistics:
     Input  bytes  :                 252
     Output bytes  :                 296
     Input  packets:                   7
     Output packets:                   8
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
     IPv6 transit statistics:
      Input  bytes  :                  0
      Output bytes  :                  0
      Input  packets:                  0
      Output packets:                  0
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 1 (last seen 00:00:00 ago)
    Output: 1 (last sent 00:00:03 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
Not-configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 1492, Generation: 171, Route table: 0
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 12.12.12.2, Local: 12.12.12.1, Broadcast: Unspecified,
Generation: 206
```


show ppp interface

Syntax show ppp interface *interface-name*
<extensive| terse>

Release Information Command introduced in Junos OS Release 7.5.

Description Display information about PPP interfaces.

Options *interface-name*—Name of a logical interface.

extensive | terse—(Optional) Display the specified level of output.

Required Privilege Level view

List of Sample Output
show ppp interface on page 153
show ppp interface extensive on page 153
show ppp interface terse on page 153

Output Fields Table 4 on page 145 lists the output fields for the **show ppp interface** command. Output fields are listed in the approximate order in which they appear.

Table 4: show ppp interface Output Fields

Field Name	Field Description	Level of Output
Session	Name of the logical interface on which the session is running.	All levels
Type	Session type: PPP.	All levels
Phase	PPP process phase: **Authenticate**, **Pending**, **Establish**, **LCP**, **Network**, **Disabled**, and **Tunneled**.	All levels
Session flags	Special conditions present in the session: **Bundled**, **TCC**, **No-keepalives**, **Looped**, **Monitored**, and **NCP-only**.	All levels
***protocol* State**	Protocol state information. See specific protocol state fields for information.	None specified
AUTHENTICATION	Challenge-Handshake Authentication Protocol (CHAP) authentication state information or Password Authentication Protocol (PAP) state information. See the **Authentication** field description for further information.	None specified

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Keepalive settings	Keepalive settings for the PPP sessions on the L2TP network server (LNS). LNS based PPP sessions are supported only on service interfaces (si). • **Interval**—Time in seconds between successive keepalive requests. Keepalive aging timeout is calculated as a product of the **interval** and **Down-count** values. If the keepalive aging timeout is greater than 180 seconds, the keepalive packets are handled by the Routing Engine. If the aging timeout is less than or equal to 180 seconds, the packets are handled by the Packet Forwarding Engine. • **Up-count**—The number of keepalive packets a destination must receive to change a link's status from down to up. • **Down-count**—The number of keepalive packets a destination must fail to receive before the network takes down a link.	**extensive**
RE Keepalive statistics	Keepalive statistics for the packets handled by the Routing Engine. • **LCP echo req Tx**—LCP echo requests sent from the Routing Engine. • **LCP echo req Rx**—LCP echo requests received at the Routing Engine. • **LCP echo rep Tx**—LCP echo responses sent from the Routing Engine. • **LCP echo rep Rx**—LCP echo responses received at the Routing Engine. • **LCP echo req timeout**—Number of keepalive packets where the keepalive aging timer has expired. • **LCP Rx echo req Magic Num Failures**—LCP echo requests where the magic numbers shared between the PPP peers during LCP negotiation did not match. • **LCP Rx echo rep Magic Num Failures**—LCP echo responses where the magic numbers shared between the PPP peers during LCP negotiation did not match.	**extensive**

Table 4: show ppp interface Output Fields *(continued)*

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Level of Output</th></tr>
<tr><td>LCP</td><td>LCP information:

State—LCP protocol state (all platforms except M120 and M320 routers):

Ack-rcvd—A Configure-Request has been sent and a Configure-Ack has been received.
Ack-sent—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
Closed—Link is not available for traffic.
Opened—Link is administratively available for traffic.
Req-sent—An attempt has been made to configure the connection.

State—LCP protocol state (M120 and M320 routers):

Ack-rcvd—A Configure-Request has been sent and a Configure-Ack has been received.
Ack-sent—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
Closed—Link is available (up), but no Open has occurred.
Closing—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
Opened—Link is administratively available for traffic. A Configure-Ack has been both sent and received.
Req-sent—An attempt has been made to configure the connection. A Configure-Request has been sent but a Configure-Ack has not yet been received.
Starting—An administrative Open has been initiated, but the lower layer is still unavailable (Down).
Stopped—The system is waiting for a Down event after the This-Layer-Finished action, or after sending a Terminate-Ack.
Stopping—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.

Last started—LCP state start time.
Last completed—LCP state completion time.
</td><td>extensive</td></tr>
</table>

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output

	• **Negotiated options**:
	  ◦ **ACFC**—Address and-Control Field Compression. A configuration option that provides a method to negotiate the compression of the Data Link Layer Address and Control fields.
	  ◦ **Asynchronous map**—Asynchronous control character map. A configuration option used on asynchronous links such as telephone lines to identify control characters that must be replaced by a two-character sequence to prevent them from being interpreted by equipment used to establish the link.
	  ◦ **Authentication protocol**—Protocol used for authentication. This option provides a method to negotiate the use of a specific protocol for authentication. It requires a peer to authenticate itself before allowing network-layer protocol packets to be exchanged. By default, authentication is not required.
	  ◦ **Authentication algorithm**—Type of authentication algorithm. The Message Digest algorithm (MD5) is the only algorithm supported.
	  ◦ **Endpoint discriminator class**—For multilink PPP (MLPPP), a configuration option that identifies the system transmitting the packet. This option advises a system that the peer on this link could be the same as the peer on another existing link.
	  ◦ **Magic number**—A configuration option that provides a method to detect looped-back links and other data-link layer anomalies. By default, the magic number is not negotiated.
	  ◦ **MRU**—Maximum receive unit. A configuration option that may be sent to inform the peer that the implementation can receive larger packets, or to request that the peer send smaller packets. The default value is 1500 octets.
	  ◦ **MRRU**—For multilink PPP, the maximum receive reconstructed unit. A configuration option that specifies the maximum number of octets in the Information fields of reassembled packets.
	  ◦ **Multilink header suspendable classes**—For MLPPP, an LCP option that advises the peer that the implementation wishes to receive fragments with a format given by the code number, with the maximum number of suspendable classes given.
	  ◦ **Multilink header format classes**—For MLPPP, an LCP option that advises the peer that the implementation wishes to receive fragments with a format given by the code number.
	  ◦ **PFC**—Protocol-Field-Compression. A configuration option that provides a method to negotiate the compression of the PPP Protocol field.
	  ◦ **short sequence**—For MLPPP, an option that advises the peer that the implementation wishes to receive fragments with short, 12-bit sequence numbers.	

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Authentication	CHAP or PAP authentication state information. For CHAP authentication: • **Chap-ans-rcvd**—Packet was sent from the peer, indicating that the peer received the **Chap-resp-sent** packet. • **Chap-ans-sent**—Packet was sent from the authenticator, indicating that the authenticator received the peer's **Chap-resp-rcvd** packet. • **Chap-chal-rcvd**—Challenge packet has been received by the peer. • **Chap-chal-sent**—Challenge packet has been sent by the authenticator to begin the CHAP protocol or has been transmitted at any time during the Network-Layer Protocol (NCP) phase to ensure that the connection has not been altered. • **Chap-resp-rcvd**—CHAP response packet has been received by the authenticator. • **Chap-resp-sent**—CHAP response packet has been sent to the authenticator. • **Closed**—Link is not available for authentication. • **Failure**—Authenticator compares the response value in the response packet from the peer with its own response value, but the value does not match. Authentication fails. • **Success**—Authenticator compares the response value in the response packet from the peer with its own response value, and the value matches. Authentication is successful. For PAP authentication: • **Pap-resp-sent**—PAP response sent to peer (ACK/NACK)t. • **Pap-req-rcvd**—PAP request packet received from peer. • **Pap-resp-rcvd**—PAP response received from the peer (ACK/NACK). • **Pap-req-sent**—PAP request packet sent to the peer. • **Closed**—Link is not available for authentication. • **Failure**—Authenticator compares the response value in the response packet from the peer with its own response value, but the value does not match. Authentication fails. • **Success**—Authenticator compares the response value in the response packet from the peer with its own response value, and the value matches. Authentication is successful.	None specified

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output

IPCP	Internet Protocol Control Protocol (IPCP) information.
	• **State**—(All platforms except M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is not available for traffic.
	  ◦ **Opened**—Link is administratively available for traffic.
	  ◦ **Req-sent**—An attempt has been made to configure the connection.
	• **State**—(M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is available (up), but no Open has occurred.
	  ◦ **Closing**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	  ◦ **Opened**—Link is administratively available for traffic. A Configure-Ack has been both sent and received.
	  ◦ **Req-sent**—An attempt has been made to configure the connection. A Configure-Request has been sent but a Configure-Ack has not yet been received.
	  ◦ **Starting**—An administrative Open has been initiated, but the lower layer is still unavailable (Down).
	  ◦ **Stopped**—The system is waiting for a Down event after the This-Layer-Finished action, or after sending a Terminate-Ack.
	  ◦ **Stopping**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	• **Last started**—IPCP state start time.
	• **Last completed**—IPCP state authentication completion time.
	• **Negotiated options:**
	  ◦ **compression protocol**—Negotiate the use of a specific compression protocol. By default, compression is not enabled.
	  ◦ **local address**—Desired local address of the sender of a Configure-Request. If all four octets are set to zero, the peer provides the IP address.
	  ◦ **primary DNS server**—Negotiate with the remote peer to select the address of the primary DNS server to be used on the local end of the link.
	  ◦ **primary WINS server**—Negotiate with the remote peer to select the address of the primary WINS server to be used on the local end of the link.
	  ◦ **remote address**—IP address of the remote end of the link in dotted quad notation.
	  ◦ **secondary DNS server**—Negotiate with the remote peer to select the address of the secondary DNS server to be used on the local end of the link.
	  ◦ **secondary WINS server**—Negotiate with the remote peer to select the address of the secondary WINS server to be used on the local end of the link.
	• **Negotiation mode**—PPP Network Control Protocol (NCP) negotiation mode configured for IPCP: **Active** or **Passive**
		extensive

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output

IPV6CP	Internet Protocol version 6 Control Protocol (IPv6CP) information.
	• **State**—(All platforms except M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is not available for traffic.
	  ◦ **Opened**—Link is administratively available for traffic.
	  ◦ **Req-sent**—An attempt has been made to configure the connection.
	• **State**—(M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is available (up), but no Open has occurred.
	  ◦ **Closing**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	  ◦ **Opened**—Link is administratively available for traffic. A Configure-Ack has been both sent and received.
	  ◦ **Req-sent**—An attempt has been made to configure the connection. A Configure-Request has been sent but a Configure-Ack has not yet been received.
	  ◦ **Starting**—An administrative Open has been initiated, but the lower layer is still unavailable (Down).
	  ◦ **Stopped**—The system is waiting for a Down event after the This-Layer-Finished action, or after sending a Terminate-Ack.
	  ◦ **Stopping**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	• **Last started**—IPV6CP state start time.
	• **Last completed**—IPV6CP state authentication completion time.
	• **Negotiated options**:
	  ◦ **local interface identifier**—Desired local address of the sender of a Configure-Request. If all four octets are set to zero, the peer provides the IP address.
	  ◦ **remote interface identifier**—IP address of the remote end of the link in dotted quad notation.
	• **Negotiation mode**—PPP Network Control Protocol (NCP) negotiation mode configured for IPv6CP: **Active** or **Passive**
		extensive

Table 4: show ppp interface Output Fields *(continued)*

Field Name	Field Description	Level of Output

OSINLCP State	OSI Network Layer Control Protocol (OSINLCP) protocol state information (all platforms except M120 and M320 routers):
	• **State:**
	  ◦ **Ack-rcvd**—Configure-Request has been sent and Configure-Ack has been received.
	  ◦ **Ack-sent**—Configure-Request and Configure-Ack have both been sent, but Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is not available for traffic.
	  ◦ **Opened**—Link is administratively available for traffic.
	  ◦ **Req-sent**—Attempt has been made to configure the connection.
	• **Last started**—OSINLCP state start time.
	• **Last completed**—OSINCLP state completion time.
		extensive

TAGCP	TAGCP information.
	• **State**—(All platforms except M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is not available for traffic.
	  ◦ **Opened**—Link is administratively available for traffic.
	  ◦ **Req-sent**—An attempt has been made to configure the connection.
	• **State**—(M120 and M320 routers) One of the following values:
	  ◦ **Ack-rcvd**—A Configure-Request has been sent and a Configure-Ack has been received.
	  ◦ **Ack-sent**—A Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received.
	  ◦ **Closed**—Link is available (up), but no Open has occurred.
	  ◦ **Closing**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	  ◦ **Opened**—Link is administratively available for traffic. A Configure-Ack has been both sent and received.
	  ◦ **Req-sent**—An attempt has been made to configure the connection. A Configure-Request has been sent but a Configure-Ack has not yet been received.
	  ◦ **Starting**—An administrative Open has been initiated, but the lower layer is still unavailable (Down).
	  ◦ **Stopped**—The system is waiting for a Down event after the This-Layer-Finished action, or after sending a Terminate-Ack.
	  ◦ **Stopping**—A Terminate-Request has been sent but a Terminate-Ack has not yet been received.
	• **Last started**—TAGCP state start time.
	• **Last completed**—TAGCP state authentication completion time.
		extensive
		none

Sample Output

show ppp interface

```
user@host> show ppp interface si-1/3/0.0
Session si-1/3/0.0, Type: PPP, Phase: Authenticate
  Session flags: Monitored
    LCP State: Opened
    AUTHENTICATION: CHAP State: Chap-resp-sent, Chap-ans-sent
    IPCP State: Closed, OSINLCP State: Closed
```

show ppp interface extensive

```
user@host> show ppp interface si-0/0/3.0 extensive

    Session si-0/0/3.0, Type: PPP, Phase: Network
        Keepalive settings: Interval 30 seconds, Up-count 1, Down-count 3
RE Keepalive statistics:
  LCP echo req Tx        :  657 (last sent 00:50:10 ago)
  LCP echo req Rx        :    0 (last seen: never)
  LCP echo rep Tx        :    0
  LCP echo rep Rx        :  657
  LCP echo req timout    :    0
  LCP Rx echo req Magic Num Failures :    0
  LCP Rx echo rep Magic Num Failures :    0
LCP
  State: Opened
  Last started: 2007-01-29 10:43:50 PST
  Last completed: 2007-01-29 10:43:50 PST
  Negotiated options:
    Authentication protocol: PAP, Magic number: 2341124815, MRU: 4470
Authentication: PAP
  State: Success
  Last started: 2007-01-29 10:43:50 PST
  Last completed: 2007-01-29 10:43:50 PST
IPCP
  State: Opened
  Last started: 2007-01-29 10:43:50 PST
  Last completed: 2007-01-29 10:43:50 PST
  Negotiated options:
    Local address: 10.10.10.1, Remote address: 10.10.10.2
  Negotiation mode: Active
IPV6CP
  State: Opened
  Last started: 2007-01-29 10:43:50 PST
  Last completed: 2007-01-29 10:43:50 PST
  Negotiated options:
    Local interface identifier: 2a0:a522:64:d319, Remote interface identifier: 0:0:0:c
  Negotiation mode: Passive
```

show ppp interface terse

```
user@host> show ppp interface si-1/3/0 terse
Session name     Session type   Session phase     Session flags
si-1/3/0.0       PPP            Authenticate      Monitored
```

show services l2tp session

Syntax

show services l2tp session
<brief | detail | extensive>
<interface *interface-name*>
<local-gateway *gateway-address*>
<local-gateway-name *gateway-name*>
<local-session-id *session-id*>
<local-tunnel-id *tunnel-id*>
<peer-gateway *gateway-address*>
<peer-gateway-name *gateway-name*>
<statistics>
<tunnel-group *group-name*>
<user *username*>

Release Information

Command introduced before Junos OS Release 7.4.
Support for LAC on MX Series routers introduced in Junos OS Release 10.4.
Support for LNS on MX Series routers introduced in Junos OS Release 11.4.

Description

(M10i and M7i routers only) Display information about active L2TP sessions for LNS.

(MX Series routers only) Display information about active L2TP sessions for LAC and LNS.

Options

none—Display standard information about all active L2TP sessions.

brief | detail | extensive—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display L2TP session information for only the specified adaptive services or inline services interface. The interface type depends on the line card as follows:

- **si-*fpc/pic/port***— MPCs on MX Series routers only. This option is not available for L2TP on M Series routers.
- **sp-*fpc/pic/port***—AS or Multiservices PICs on M7i, M10i, and M120 routers only. This option is not available for L2TP on MX Series routers.

local-gateway *gateway-address*—(Optional) Display L2TP session information for only the specified local gateway address.

local-gateway-name *gateway-name*—(Optional) Display L2TP session information for only the specified local gateway name.

local-session-id *session-id*—(Optional) Display L2TP session information for only the specified local session identifier.

local-tunnel-id *tunnel-id*—(Optional) Display L2TP session information for only the specified local tunnel identifier.

peer-gateway *gateway-address*—(Optional) Display L2TP session information for only the specified peer gateway address.

peer-gateway-name ***gateway-name***—(Optional) Display L2TP session information for only the specified peer gateway name.

statistics—(Optional) Display the number of control packets and bytes transmitted and received for the session. You cannot include this option with any of the level options, **brief**, **detail**, or **extensive**.

tunnel-group ***group-name***—(Optional) Display L2TP session information for only the specified tunnel group. To display information about L2TP CPU and memory usage, you can include the tunnel group name in the **show services service-sets memory-usage** ***group-name*** and **show services service-sets cpu-usage** ***group-name*** commands. This option is not available for L2TP LAC on MX Series routers.

user ***username***—(M Series routers only) (Optional) Display L2TP session information for only the specified username.

Required Privilege Level view

Related Documentation

- *L2TP Services Configuration Overview*
- *L2TP Minimum Configuration*
- *clear services l2tp session*

List of Sample Output

Output Fields Table 5 on page 155 lists the output fields for the **show services l2tp session** command. Output fields are listed in the approximate order in which they appear.

Table 5: show services l2tp session Output Fields

Field Name	Field Description	Level of Output
Interface	(LNS only) Name of an adaptive services interface.	All levels
Tunnel group	(LNS only) Name of a tunnel group.	All levels
Tunnel local ID	Identifier of the local endpoint of the tunnel, as assigned by the L2TP network server (LNS).	All levels
Session local ID	Identifier of the local endpoint of the L2TP session, as assigned by the LNS.	All levels

Table 5: show services l2tp session Output Fields *(continued)*

Field Name	Field Description	Level of Output
Session remote ID	Identifier of the remote endpoint of the L2TP session, as assigned by the L2TP access concentrator (LAC).	All levels
State	State of the L2TP session: • **Established**—Session is operating. This is the only state supported for the LAC. • **closed**—Session is being closed. • **destroyed**—Session is being destroyed. • **clean-up**—Session is being cleaned up. • **lns-ic-accept-new**—New session is being accepted. • **lns-ic-idle**—Session has been created and is idle. • **lns-ic-reject-new**—New session is being rejected. • **lns-ic-wait-connect**—Session is waiting for the peer's incoming call connected (ICCN) message.	All levels
Bundle ID	(LNS only) Bundle identifier. Indicates the session is part of a multilink bundle. Sessions that have a blank **Bundle** field are not participating in the Multilink Protocol. Sessions in a multilink bundle might belong to different L2TP tunnels. For L2TP output organized by bundle ID, issue the **show services l2tp multilink extensive** command.	All levels
Mode	(LNS) Mode of the interface representing the session: **shared** or **exclusive**. (LAC) Mode of the interface representing the session: **shared** or **dedicated**. Only **dedicated** is currently supported for the LAC.	**extensive**
Local IP	IP address of local endpoint of the Point-to-Point Protocol (PPP) session.	**extensive**
Remote IP	IP address of remote endpoint of the PPP session.	**extensive**
Username	(LNS only) Name of the user logged in to the session.	All levels
Assigned IP address	(LNS only) IP address assigned to remote client.	**extensive**
Local name	For LNS, name of the LNS instance in which the session was created. For LAC, name of the LAC.	**extensive**
Remote name	For LNS, name of the LAC from which the session was created. For LAC, name of the LAC instance.	**extensive**
Local MRU	(LNS only) Maximum receive unit (MRU) setting of the local device, in bytes.	**extensive**
Remote MRU	(LNS only) MRU setting of the remote device, in bytes.	**extensive**
Tx speed	Transmit speed of the session conveyed from the LAC to the LNS, in bits per second (bps). Both the initial (**initial**) and current (**update**) line speeds can be displayed on MX Series routers.	**extensive**

Table 5: show services l2tp session Output Fields *(continued)*

Field Name	Field Description	Level of Output
Rx speed	Receive speed of the session conveyed from the LAC to the LNS, in bits per second (bps). Both the initial (**initial**) and current (**update**) line speeds can be displayed on MX Series routers.	**extensive**
Bearer type	Type of bearer enabled: • **0**—Might indicate that the call was not received over a physical link (for example, when the LAC and PPP are located in the same subsystem). • **1**—Digital access requested. • **2**—Analog access requested. • **4**—Asynchronous Transfer Mode (ATM) bearer support.	**extensive**
Framing type	Type of framing enabled: • **1**—Synchronous framing • **2**—Asynchronous framing	**extensive**
LCP renegotiation	(LNS only) Whether Link Control Protocol (LCP) renegotiation is configured: **On** or **Off**.	**extensive**
Authentication	Type of authentication algorithm used: Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP).	**extensive**
Interface ID	(LNS only) Identifier used to look up the logical interface for this session.	**extensive**
Interface unit	Logical interface for this session.	All levels
Call serial number	Unique serial number assigned to the call.	**extensive**
Policer bandwidth	Maximum policer bandwidth configured for this session.	**extensive**
Policer burst size	Maximum policer burst size configured for this session.	**extensive**
Firewall filter	Configured firewall filter name.	**extensive**
Session encapsulation overhead	Overhead allowance configured for this session, in bytes.	**extensive**
Session cell overhead	Cell overhead activation (**On** or **Off**).	**extensive**
Create time	Date and time when the call was created.	**extensive**
Up time	Length of time elapsed since the call became active, in hours, minutes, and seconds.	**extensive**

Table 5: show services l2tp session Output Fields *(continued)*

Field Name	Field Description	Level of Output
Idle time	Length of time elapsed since the call became idle, in hours, minutes, and seconds.	**extensive**
Statistics since	Date and time when collection of the following statistics began: • **Control Tx**—Amount of control information transmitted, in packets and bytes. • **Control Rx**—Amount of control information received, in packets and bytes. • **Data Tx**—Amount of data transmitted, in packets and bytes. • **Data Rx**—Amount of data received, in packets and bytes. • **Errors Tx**—Number of errors transmitted, in packets. • **Errors Rx**—Number of errors received, in packets. • **LCP echo req Tx**—Number of LCP echo requests transmitted, in packets. • **LCP echo req Rx**—Number of LCP echo requests received, in packets. • **LCP echo rep Tx**—Number of LCP echo responses transmitted, in packets. • **LCP echo rep Rx**—Number of LCP echo responses received, in packets. • **LCP echo Req timout**—Number of LCP echo requests that timed out. • **LCP echo Req error**—Number of errors received for LCP echo packets. • **LCP echo Rep error** —Number of errors transmitted for LCP echo packets.	**extensive**

Sample Output

show services l2tp session (LNS on M Series Routers)

```
user@host> show services l2tp session
Interface: sp-1/2/0, Tunnel group: group1, Tunnel local ID:  8802
  Local Remote Interface State               Bundle Username
  ID    ID     unit
  37966      5         2 Established
```

show services l2tp session (LNS on MX Series Routers)

```
user@host> show services l2tp session
Tunnel local ID: 40553
  Local   Remote   State                Interface         Interface
  ID      ID                            unit              Name
  17967   1        Established          1073749824        si-5/2/0
```

show services l2tp session (LAC)

```
user@host> show services l2tp session
Tunnel local ID: 31889
  Local   Remote   State                Interface         Interface
  ID      ID                            unit              Name
   31694      1    Established                311         pp0
```

show services l2tp session detail (LAC)

```
user@host> show services l2tp session detail
Tunnel local ID: 31889
  Session local ID: 31694, Session remote ID:     1, Interface unit: 311
    State: Established, Interface: pp0, Mode: Dedicated
```

```
    Local IP: 10.1.1.2:1701, Remote IP: 10.1.1.1:1701
    Local name: ce-lac, Remote name: ce-lns
```

show services l2tp session extensive (LAC)

```
user@host> show services l2tp session extensive
Tunnel local ID: 31889
  Session local ID: 31694, Session remote ID:     1
    Interface unit: 311
    State: Established, Mode: Dedicated
    Local IP: 10.10.1.2:1701, Remote IP: 10.10.1.1:1701
    Local name: ce-lac, Remote name: ce-lns
    Tx speed: 0, Rx speed: 0
    Bearer type: 1, Framing type: 1
    LCP renegotiation: N/A, Authentication: None, Interface ID: N/A
    Interface unit: 311, Call serial number: 0
    Policer bandwidth: 0, Policer burst size: 0
    Policer exclude bandwidth: 0, Firewall filter: 0
    Session encapsulation overhead: 0, Session cell overhead: 0
    Create time: Tue Aug 24 14:38:23 2010, Up time: 01:06:25
    Idle time: N/A
```

show services l2tp session extensive (LAC on MX Series Routers)

```
user@host> show services l2tp session extensive
Tunnel local ID: 31889
  Session local ID: 31694, Session remote ID:     1
    Interface unit: 311
    State: Established, Mode: Dedicated
    Local IP: 10.10.1.2:1701, Remote IP: 10.10.1.1:1701
    Local name: ce-lac, Remote name: ce-lns
    Tx speed: initial 64000, Update 256000
    Rx speed: initial 64000, Update 256000
    Bearer type: 1, Framing type: 1
    LCP renegotiation: N/A, Authentication: None, Interface ID: N/A
    Interface unit: 311, Call serial number: 0
    Policer bandwidth: 0, Policer burst size: 0
    Policer exclude bandwidth: 0, Firewall filter: 0
    Session encapsulation overhead: 0, Session cell overhead: 0
    Create time: Tue Aug 24 14:38:23 2010, Up time: 01:06:25
    Idle time: N/A
```

show services l2tp session extensive (LNS on M Series Routers)

```
user@host> show services l2tp session extensive
Interface: sp-1/2/0, Tunnel group: group1, Tunnel local ID: 62746
  Session local ID: 56793, Session remote ID: 53304
    State: Established, Bundle ID: 5, Mode: shared
    Local IP: 10.128.1.1:1701, Remote IP: 10.128.1.2:1701
    Username: usr1@juniper_1.net, Assigned IP address: 10.50.2.1/32
    Local MRU: 4000, Remote MRU: 1500, Tx speed: 64000, Rx speed: 64000
    Bearer type: 2, Framing type: 1
    LCP renegotiation: Off, Authentication: CHAP, Interface ID: unit_20
    Interface unit: 20, Call serial number: 4137941434
    Policer bandwidth: 64000, Policer burst size: 51200
    Firewall filter: f1
    Session encapsulation overhead: 16, Session cell overhead: On
    Create time: Tue Mar 23 14:13:15 2004, Up time: 01:16:41
    Idle time: 00:00:00
    Statistics since: Tue Mar 23 14:13:13 2004
                        Packets        Bytes
      Control Tx              4           88
```



```
      Control Rx              2           28
      Data Tx                 0            0
      Data Rx               461        29.0k
      Errors Tx               0
      Errors Rx               0

Interface: sp-1/2/0, Tunnel group: group_company_dns, Tunnel local ID: 37266
  Session local ID: 39962, Session remote ID: 53303
    State: Established, Bundle ID: 5, Mode: shared
    Local IP: 10.128.11.1:1701, Remote IP: 10.128.11.2:1701
    Username: usr1@company.com, Assigned IP address: 10.46.2.3/24
    Local name: router-1, Remote name: router-2
    Local MRU: 4470, Remote MRU: 4470, Tx speed: 155000000, Rx speed: 155000000
    Bearer type: 2, Framing type: 1
    LCP renegotiation: Off, Authentication: CHAP, Interface ID: unit_31
    Interface unit: 31, Call serial number: 4137941433
    Policer bandwidth: 64000, Policer burst size: 51200
    Firewall filter: f1
    Create time: Tue Mar 23 14:13:17 2004, Up time: 01:16:39
    Idle time: 01:16:36
    Statistics since: Tue Mar 23 14:13:15 2004
                        Packets        Bytes
      Control Tx              6          196
      Control Rx              4          150
      Data Tx                 0            0
      Data Rx                 1           80
      Errors Tx               0
      Errors Rx               0
```

show services l2tp session extensive (LNS on MX Series Routers)

```
user@host> show services l2tp session extensive
Tunnel local ID: 40553
  Session local ID: 17967, Session remote ID: 1
    Interface unit: 1073749824
    State: Established
    Interface: si-5/2/0
    Mode: Dedicated
    Local IP: 11.1.1.2:1701, Remote IP: 11.1.1.3:1701
    Local name: lns-mx960, Remote name: testlac
    Tx speed: 56000, Rx speed: 0
    Bearer type: 2, Framing type: 1
    LCP renegotiation: Off, Authentication: None
    Call serial number: 1
    Create time: Mon Apr 25 20:27:50 2011, Up time: 00:01:48
    Idle time: N/A
    Statistics since: Mon Apr 25 20:27:50 2011
                        Packets        Bytes
      Control Tx              4          219
      Control Rx              4          221
      Data Tx                 0            0
      Data Rx                10          228
      Errors Tx               0
      Errors Rx               0
```

show services l2tp session statistics (MX Series Routers)

```
user@host>show services l2tp session statistics local session-id 1
Tunnel local ID: 17185
Session local ID: 1, Session remote ID: 14444, Interface unit: 1073788352
State: Established
```



```
Statistics since: Mon Aug 1 13:27:47 2011
    Packets   Bytes
Data Tx   4    51
Data Rx   3    36
```

show subscribers

Syntax

show subscribers
<detail | extensive | terse>
<aci-interface-set-name *aci-interface-set-name*>
<address *address*>
<agent-circuit-identifier *agent-circuit-identifier-substring*>
<client-type *client-type*>
<count>
<id>
<interface *interface*>
<logical-system *logical-system*>
<mac-address *mac-address*>
<physical-interface *physical-interface-name*>
<profile-name *profile-name*>
<routing-instance *routing-instance*>
<stacked-vlan-id *stacked-vlan-id*>
<subscriber-state *subscriber-state*>
<user-name *user-name*>
<vci *vci-identifier*>
<vpi *vpi-identifier*>
<vlan-id *vlan-id*>

Release Information

Command introduced in Junos OS Release 9.3.
Command introduced in Junos OS Release 9.3 for EX Series switches.
client-type, **mac-address**, **subscriber-state**, and **extensive** options introduced in Junos OS Release 10.2.
count option usage with other options introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Options **aci-interface-set-name** and **agent-circuit-identifier** introduced in Junos OS Release 12.2.
The **physical-interface** and **user-name** options introduced in Junos OS Release 12.3.
Options **vci** and **vpi** introduced in Junos OS Release 12.3R3 and supported in later 12.3R*x* releases.
Options **vci** and **vpi** supported in Junos OS Release 13.2 and later releases. (Not supported in Junos OS Release 13.1.)

Description

Display information for active subscribers.

Options

detail | extensive | terse—(Optional) Display the specified level of output.

aci-interface-set-name—(Optional) Display all dynamic subscriber sessions that use the specified agent circuit identifier (ACI) interface set. Use the ACI interface set name generated by the router, such as aci-1003-ge-1/0/0.4001, and not the actual ACI value found in the DHCP or PPPoE control packets.

address—(Optional) Display subscribers whose IP address matches the specified address. You must specify the IPv4 or IPv6 address prefix without a netmask (for example, 192.168.17.1). If you specify the IP address as a prefix with a netmask (for example, 192.168.17.1/32), the router displays a message that the IP address is invalid, and rejects the command.

agent-circuit-identifier-substring—(Optional) Display all dynamic subscriber sessions whose ACI value matches the specified substring.

client-type—(Optional) Display subscribers whose client type matches the specified client type (DHCP, L2TP, PPP, PPPOE, VLAN, or static).

count—(Optional) Display the count of total subscribers and active subscribers for any specified option. You can use the **count** option alone or with the **address**, **client-type**, **interface**, **logical-system**, **mac-address**, **profile-name**, **routing-instance**, **stacked-vlan-id**, **subscriber-state**, or **vlan-id** options.

id—(Optional) Display a specific subscriber session whose session id matches the specified subscriber ID. You can display subscriber IDs by using the **show subscribers extensive** or the **show subscribers interface extensive** commands.

interface—(Optional) Display subscribers whose interface matches the specified interface.

logical-system—(Optional) Display subscribers whose logical system matches the specified logical system.

mac-address—(Optional) Display subscribers whose MAC address matches the specified MAC address.

physical-interface-name—(M120, M320, and MX Series routers only) (Optional) Display subscribers whose physical interface matches the specified physical interface.

profile-name—(Optional) Display subscribers whose dynamic profile matches the specified profile name.

routing-instance—(Optional) Display subscribers whose routing instance matches the specified routing instance.

stacked-vlan-id—(Optional) Display subscribers whose stacked VLAN ID matches the specified stacked VLAN ID.

subscriber-state—(Optional) Display subscribers whose subscriber state matches the specified subscriber state (ACTIVE, CONFIGURED, INIT, TERMINATED, or TERMINATING).

user-name—(M120, M320, and MX Series routers only) (Optional) Display subscribers whose username matches the specified subscriber name.

vci-identifier—(MX Series routers with MPCs and ATM MICs with SFP only) (Optional) Display active ATM subscribers whose ATM virtual circuit identifier (VCI) matches the specified VCI identifier. The range of values is **0** through **255**.

vpi-identifier—(MX Series routers with MPCs and ATM MICs with SFP only) (Optional) Display active ATM subscribers whose ATM virtual path identifier (VPI) matches the specified VPI identifier. The range of values is **0** through **65535**.

vlan-id—(Optional) Display subscribers whose VLAN ID matches the specified VLAN ID.

NOTE: Due to display limitations, logical system and routing instance output values are truncated when necessary.

Required Privilege Level

view

Related Documentation

- show subscribers summary on page 180
- *Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration*

List of Sample Output

show subscribers vlan-id detail on page 179
show subscribers vpi vci extensive (PPPoE-over-ATM Subscriber Session) on page 179

Output Fields Table 6 on page 165 lists the output fields for the **show subscribers** command. Output fields are listed in the approximate order in which they appear.

Table 6: show subscribers Output Fields

Field Name	Field Description
Interface	Interface associated with the subscriber. The router or switch displays subscribers whose interface matches or begins with the specified interface. The * character indicates a continuation of addresses for the same session.
IP Address/VLAN ID	Subscriber IP address or VLAN ID associated with the subscriber in the form *tpid.vlan-id* No IP address or VLAN ID is assigned to an L2TP tunnel-switched session. For these subscriber sessions the value is **Tunnel-switched**.
User Name	Name of subscriber.
LS:RI	Logical system and routing instance associated with the subscriber.
Type	Subscriber client type (DHCP, L2TP, PPP, PPPoE, STATIC-INTERFACE, VLAN).
IP Address	Subscriber IPv4 address.
IP Netmask	Subscriber IP netmask.
Primary DNS Address	IP address of primary DNS server.
Secondary DNS Address	IP address of secondary DNS server.
Primary WINS Address	IP address of primary WINS server.
Secondary WINS Address	IP address of secondary WINS server.
IPv6 Address	Subscriber IPv6 address, or multiple addresses.
IPv6 Prefix	Subscriber IPv6 prefix. If you are using DHCPv6 prefix delegation, this is the delegated prefix.
IPv6 User Prefix	IPv6 prefix obtained through ND/RA.
IPv6 Address Pool	Subscriber IPv6 address pool. The IPv6 address pool is used to allocate IPv6 prefixes to the DHCPv6 clients.
IPv6 Network Prefix Length	Length of the network portion of the IPv6 address.
IPv6 Prefix Length	Length of the subscriber IPv6 prefix.

Table 6: show subscribers Output Fields *(continued)*

Field Name	Field Description
Logical System	Logical system associated with the subscriber.
Routing Instance	Routing instance associated with the subscriber.
Interface Type	Whether the subscriber interface is **Static** or **Dynamic**.
Interface Set	Internally generated name of the dynamic ACI interface set used by the subscriber session.
Interface Set Type	Interface type of the ACI interface set: **Dynamic**. This is the only ACI interface set type currently supported.
Interface Set Session ID	Identifier of the dynamic ACI interface set entry in the session database.
Underlying Interface	Name of the underlying interface for the subscriber session.
Dynamic Profile Name	Dynamic profile used for the subscriber.
Dynamic Profile Version	Version number of the dynamic profile used for the subscriber.
MAC Address	MAC address associated with the subscriber.
State	Current state of the subscriber session (**Init**, **Configured**, **Active**, **Terminating**, **Tunneled**).
L2TP State	Current state of the L2TP session, **Tunneled** or **Tunnel-switched**. When the value is **Tunnel-switched**, two entries are displayed for the subscriber; the first entry is at the LNS interface on the LTS and the second entry is at the LAC interface on the LTS.
Tunnel switch Profile Name	Name of the L2TP tunnel switch profile that initiates tunnel switching.
Local IP Address	IP address of the local gateway (LAC).
Remote IP Address	IP address of the remote peer (LNS).
VLAN Id	VLAN ID associated with the subscriber in the form *tpid.vlan-id*.
Stacked VLAN Id	Stacked VLAN ID associated with the subscriber in the form *tpid.vlan-id*.
RADIUS Accounting ID	RADIUS accounting ID associated with the subscriber.
Agent Circuit ID	Option 82 agent circuit ID associated with the subscriber. The ID is displayed as an ASCII string unless the value has nonprintable characters, in which case it is displayed in hexadecimal format.
Agent Remote ID	Option 82 agent remote ID associated with the subscriber. The ID is displayed as an ASCII string unless the value has nonprintable characters, in which case it is displayed in hexadecimal format.
DHCP Relay IP Address	IP address used by the DHCP relay agent.

Table 6: show subscribers Output Fields *(continued)*

Field Name	Field Description
ATM VPI	(MX Series routers with MPCs and ATM MICs with SFP only) ATM virtual path identifier (VPI) on the subscriber's physical interface.
ATM VCI	(MX Series routers with MPCs and ATM MICs with SFP only) ATM virtual circuit identifier (VCI) for each VPI configured on the subscriber interface.
Login Time	Date and time at which the subscriber logged in.
Effective shaping-rate	Actual downstream traffic shaping rate for the subscriber, in kilobits per second.
IPv4 rpf-check Fail Filter Name	Name of the filter applied by the dynamic profile to IPv4 packets that fail the RPF check.
IPv6 rpf-check Fail Filter Name	Name of the filter applied by the dynamic profile to IPv6 packets that fail the RPF check.
DHCP Options	**len** = number of hex values in the message. The hex values specify the type, length, value (TLV) for DHCP options, as defined in RFC 2132.
Session ID	ID number for a subscriber service session.
Underlying Session ID	For DHCPv6 subscribers on a PPPoE network, displays the session ID of the underlying PPPoE interface.
Service Sessions	Number of service sessions (that is, a service activated using RADIUS CoA) associated with the subscribers.
Service Session Name	Service session profile name.
Session Timeout (seconds)	Number of seconds of access provided to the subscriber before the session is automatically terminated.
Idle Timeout (seconds)	Number of seconds subscriber can be idle before the session is automatically terminated.
IPv6 Delegated Address Pool	Name of the pool used for DHCPv6 prefix delegation.
IPv6 Delegated Network Prefix Length	Length of the prefix configured for the IPv6 delegated address pool.
IPv6 Interface Address	Address assigned by the Framed-Ipv6-Prefix AAA attribute.
IPv6 Framed Interface Id	Interface ID assigned by the Framed-Interface-Id AAA attribute.
ADF IPv4 Input Filter Name	Name assigned to the Ascend-Data-Filter (ADF) interface IPv4 input filter (client or service session). The filter name is followed by the rules (in hexadecimal format) associated with the ADF filter and the decoded rule in Junos OS filter style.

Table 6: show subscribers Output Fields *(continued)*

Field Name	Field Description
ADF IPv4 Output Filter Name	Name assigned to the Ascend-Data-Filter (ADF) interface IPv4 output filter (client or service session). The filter name is followed by the rules (in hexadecimal format) associated with the ADF filter and the decoded rule in Junos OS filter style.
ADF IPv6 Input Filter Name	Name assigned to the Ascend-Data-Filter (ADF) interface IPv6 input filter (client or service session). The filter name is followed by the rules (in hexadecimal format) associated with the ADF filter and the decoded rule in Junos OS filter style.
ADF IPv6 Output Filter Name	Name assigned to the Ascend-Data-Filter (ADF) interface IPv6 output filter (client or service session). The filter name is followed by the rules (in hexadecimal format) associated with the ADF filter and the decoded rule in Junos OS filter style.
IPv4 Input Filter Name	Name assigned to the IPv4 input filter (client or service session).
IPv4 Output Filter Name	Name assigned to the IPv4 output filter (client or service session).
IPv6 Input Filter Name	Name assigned to the IPv6 input filter (client or service session).
IPv6 Output Filter Name	Name assigned to the IPv6 output filter (client or service session).
IFL Input Filter Name	Name assigned to the logical interface input filter (client or service session).
IFL Output Filter Name	Name assigned to the logical interface output filter (client or service session).

Sample Output

show subscribers (IPv4)

```
user@host> show subscribers
Interface               IP Address/VLAN ID   User Name         LS:RI
ge-1/3/0.1073741824     100                                    default:default
demux0.1073741824       100.0.0.10           WHOLESALER-CLIENT default:default
demux0.1073741825       101.0.0.3            RETAILER1-CLIENT  test1:retailer1
demux0.1073741826       102.0.0.3            RETAILER2-CLIENT  test1:retailer2
```

show subscribers (IPv6)

```
user@host> show subscribers
Interface          IP Address/VLAN ID   User Name           LS:RI
ge-1/0/0.0         2001::c0:0:0:0/74    WHOLESALER-CLIENT   default:default
     *             2002::1/128          subscriber-25       default:default
```

show subscribers (IPv4 and IPv6 Dual Stack)

```
user@host> show subscribers
Interface           IP Address/VLAN ID           User Name
LS:RI
demux0.1073741834   0x8100.1002 0x8100.1
default:default
demux0.1073741835   0x8100.1001 0x8100.1
default:default
pp0.1073741836      61.1.1.1                   dualstackuser1@ISP1.com
```



```
default:ASP-1
*                   2041:1:1::/48
*                   2061:1:1:1::/64
pp0.1073741837      23.1.1.3                 dualstackuser2@ISP1.com
default:ASP-1
*                   2001:1:2:5::/64
```

show subscribers (LNS on MX Series Routers)

```
user@host> show subscribers
Interface          IP Address/VLAN ID   User Name         LS:RI
si-4/0/0.1         192.168.4.1          xyz@example.com   default:default
```

show subscribers (L2TP Switched Tunnels)

```
user@host> show subscribers
Interface           IP Address/VLAN ID    User Name              LS:RI
si-2/1/0.1073741842 Tunnel-switched       ap@lts.com        default:default

si-2/1/0.1073741843 Tunnel-switched       ap@lts.com        default:default
```

show subscribers client-type dhcp detail

```
user@host> show subscribers client-type dhcp detail
Type: DHCP
IP Address: 100.20.9.7
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: demux0.1073744127
Interface type: Dynamic
Dynamic Profile Name: dhcp-demux-prof
MAC Address: 00:10:95:00:00:98
State: Active
Radius Accounting ID: jnpr :2304
Login Time: 2009-08-25 14:43:52 PDT

Type: DHCP
IP Address: 100.20.10.7
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: demux0.1073744383
Interface type: Dynamic
Dynamic Profile Name: dhcp-demux-prof
MAC Address: 00:10:94:00:01:f3
State: Active
Radius Accounting ID: jnpr :2560
Login Time: 2009-08-25 14:43:56 PDT
```

show subscribers count

```
user@host> show subscribers count
Total Subscribers: 188, Active Subscribers: 188
```

show subscribers address detail (IPv6)

```
user@host> show subscribers address 100.16.12.137 detail
```

```
Type: PPPoE
User Name: pppoeTerV6User1Svc
IP Address: 100.16.12.137
IP Netmask: 255.0.0.0
IPv6 User Prefix: 1016:0:0:c88::/64
Logical System: default
Routing Instance: default
Interface: pp0.1073745151
Interface type: Dynamic
Underlying Interface: demux0.8201
Dynamic Profile Name: pppoe-client-profile
MAC Address: 00:0d:02:01:00:01
Session Timeout (seconds): 31622400
Idle Timeout (seconds): 86400
State: Active
Radius Accounting ID: jnpr demux0.8201:6544
Session ID: 6544
Agent Circuit ID: ifl3720
Agent Remote ID: ifl3720
Login Time: 2012-05-21 13:37:27 PDT
Service Sessions: 1
```

show subscribers detail (IPv4)

```
user@host> show subscribers detail
Type: DHCP
IP Address: 100.20.9.7
IP Netmask: 255.255.0.0
Primary DNS Address: 192.168.17.1
Secondary DNS Address: 192.168.17.2
Primary WINS Address: 192.168.22.1
Secondary WINS Address: 192.168.22.2
Logical System: default
Routing Instance: default
Interface: demux0.1073744127
Interface type: Dynamic
Dynamic Profile Name: dhcp-demux-prof
MAC Address: 00:10:95:00:00:98
State: Active
Radius Accounting ID: jnpr :2304
Idle Timeout (seconds): 600
Login Time: 2009-08-25 14:43:52 PDT
DHCP Options: len 52
35 01 01 39 02 02 40 3d 07 01 00 10 94 00 00 08 33 04 00 00
00 3c 0c 15 63 6c 69 65 6e 74 5f 50 6f 72 74 20 2f 2f 36 2f
33 2d 37 2d 30 37 05 01 06 0f 21 2c
Service Sessions: 2
```

show subscribers detail (IPv6)

```
user@host> show subscribers detail
Type: DHCP
User Name: pd-user1
IPv6 Prefix: 2002:db2:ffff:1::/64
Logical System: default
Routing Instance: default
Interface: ge-3/1/3.2
Interface type: Static
MAC Address: 00:51:ff:ff:00:03
State: Active
Radius Accounting ID: 1
```



```
Session ID: 1
Login Time: 2011-08-25 12:12:26 PDT
DHCP Options: len 42
00 08 00 02 00 00 00 01 00 0a 00 03 00 01 00 51 ff ff 00 03
00 06 00 02 00 19 00 19 00 0c 00 00 00 00 00 00 00 00 00 00
00 00
```

show subscribers detail (IPv6 Static Demux Interface)

```
user@host> show subscribers detail
Type: STATIC-INTERFACE
User Name: demux0.1@jnpr.net
IPv6 Prefix: 1:2:3:4:5:6:7:aa/128
Logical System: default
Routing Instance: default
Interface: demux0.1
Interface type: Static
Dynamic Profile Name: junos-default-profile
State: Active
Radius Accounting ID: 185
Login Time: 2010-05-18 14:33:56 EDT
```

show subscribers detail (L2TP LNS Subscribers on MX Series Routers)

```
user@host> show subscribers detail
Type: L2TP
User Name: user1@jnpr.net
IP Address: 10.1.32.58
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: si-5/2/0.1073749824
Interface type: Dynamic
Dynamic Profile Name: dyn-lns-profile2
Dynamic Profile Version: 1
State: Active
Radius Accounting ID: 8001
Session ID: 8001
Login Time: 2011-04-25 20:27:50 IST
```

show subscribers detail (L2TP Switched Tunnels)

```
user@host> show subscribers detail
Type: L2TP
User Name: ap@example.com
Logical System: default
Routing Instance: default
Interface: si-2/1/0.1073741842
Interface type: Dynamic
Dynamic Profile Name: dyn-lts-profile
State: Active
L2TP State: Tunnel-switched
Tunnel switch Profile Name: ce-lts-profile
Local IP Address: 10.50.1.1
Remote IP Address: 192.168.20.3
Radius Accounting ID: 21
Session ID: 21
Login Time: 2013-01-18 03:01:11 PST

Type: L2TP
User Name: ap@example.com
Logical System: default
```

```
Routing Instance: default
Interface: si-2/1/0.1073741843
Interface type: Dynamic
Dynamic Profile Name: dyn-lts-profile
State: Active
L2TP State: Tunnel-switched
Tunnel switch Profile Name: ce-lts-profile
Local IP Address: 10.30.1.1
Remote IP Address: 172.20.1.10
Session ID: 22
Login Time: 2013-01-18 03:01:14 PST
```

show subscribers detail (Tunneled Subscriber)

```
user@host> show subscribers detail
Type: PPPoE
User Name: user1@example.com
Logical System: default
Routing Instance: default
Interface: pp0.1
State: Active, Tunneled
Radius Accounting ID: 512
```

show subscribers detail (IPv4 and IPv6 Dual Stack)

```
user@host> show subscribers detail
Type: VLAN
Logical System: default
Routing Instance: default
Interface: demux0.1073741824
Interface type: Dynamic
Dynamic Profile Name: svlanProfile
State: Active
Session ID: 1
Stacked VLAN Id: 0x8100.1001
VLAN Id: 0x8100.1
Login Time: 2011-11-30 00:18:04 PST

Type: PPPoE
User Name: dualstackuser1@ISP1.com
IP Address: 61.1.1.1
IPv6 Prefix: 2041:1:1::/48
IPv6 User Prefix: 2061:1:1:1::/64
Logical System: default
Routing Instance: ASP-1
Interface: pp0.1073741825
Interface type: Dynamic
Dynamic Profile Name: dualStack-Profile1
MAC Address: 00:00:64:03:01:02
State: Active
Radius Accounting ID: 2
Session ID: 2
Login Time: 2011-11-30 00:18:05 PST

Type: DHCP
IPv6 Prefix: 2041:1:1::/48
Logical System: default
Routing Instance: ASP-1
Interface: pp0.1073741825
Interface type: Static
MAC Address: 00:00:64:03:01:02
```



```
State: Active
Radius Accounting ID: jnpr :3
Session ID: 3
Underlying Session ID: 2
Login Time: 2011-11-30 00:18:35 PST
DHCP Options: len 42
00 08 00 02 0b b8 00 01 00 0a 00 03 00 01 00 00 64 03 01 02
00 06 00 02 00 19 00 19 00 0c 00 00 00 00 00 00 00 00 00 00
00 00
```

show subscribers detail (ACI Interface Set Session)

```
user@host> show subscribers detail
Type: VLAN
Logical System: default
Routing Instance: default
Interface: ge-1/0/0
Interface Set: aci-1001-ge-1/0/0.2800
Interface Set Session ID: 0
Underlying Interface: ge-1/0/0.2800
Dynamic Profile Name: aci-vlan-set-profile-2
Dynamic Profile Version: 1
State: Active
Session ID: 1
Agent Circuit ID: aci-ppp-dhcp-20
Login Time: 2012-05-26 01:54:08 PDT
```

show subscribers detail (PPPoE Subscriber Session with ACI Interface Set)

```
user@host> show subscribers detail
Type: PPPoE
User Name: ppphint2
IP Address: 10.10.1.5
Logical System: default
Routing Instance: default
Interface: pp0.1073741825
Interface type: Dynamic
Interface Set: aci-1001-demux0.1073741824
Interface Set Type: Dynamic
Interface Set Session ID: 2
Underlying Interface: demux0.1073741824
Dynamic Profile Name: aci-vlan-pppoe-profile
Dynamic Profile Version: 1
MAC Address: 00:00:64:39:01:02
State: Active
Radius Accounting ID: 3
Session ID: 3
Agent Circuit ID: aci-ppp-dhcp-dvlan-50
Login Time: 2012-03-07 13:46:53 PST
```

show subscribers extensive

```
user@host> show subscribers extensive
Type: DHCP
User Name: pd-user1
IPv6 Prefix: 2002:db2:ffff:1::/64
Logical System: default
Routing Instance: default
Interface: ge-3/1/3.2
Interface type: Static
MAC Address: 00:51:ff:ff:00:03
```



```
State: Active
Radius Accounting ID: 1
Session ID: 1
Login Time: 2011-08-25 12:12:26 PDT
DHCP Options: len 42
00 08 00 02 00 00 00 01 00 0a 00 03 00 01 00 51 ff ff 00 03
00 06 00 02 00 19 00 19 00 0c 00 00 00 00 00 00 00 00 00 00
00 00
IPv6 Address Pool: pd_pool
IPv6 Network Prefix Length: 48
```

show subscribers extensive (RPF Check Fail Filter)

```
user@host> show subscribers extensive
...
Type: VLAN
  Logical System: default
  Routing Instance: default
  Interface: ae0.1073741824
  Interface type: Dynamic
  Dynamic Profile Name: vlan-prof
  State: Active
  Session ID: 9
  VLAN Id: 100
  Login Time: 2011-08-26 08:17:00 PDT
  IPv4 rpf-check Fail Filter Name: rpf-allow-dhcp
  IPv6 rpf-check Fail Filter Name: rpf-allow-dhcpv6
...
```

show subscribers extensive (L2TP LNS Subscribers on MX Series Routers)

```
user@host> show subscribers extensive
Type: L2TP
User Name: user1@jnpr.net
IP Address: 10.1.32.58
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: si-5/2/0.1073749824
Interface type: Dynamic
Dynamic Profile Name: dyn-lns-profile2
Dynamic Profile Version: 1
State: Active
Radius Accounting ID: 8001
Session ID: 8001
Login Time: 2011-04-25 20:27:50 IST
IPv4 Input Filter Name: classify-si-5/2/0.1073749824-in
IPv4 Output Filter Name: classify-si-5/2/0.1073749824-out
```

show subscribers extensive (IPv4 and IPv6 Dual Stack)

```
user@host> show subscribers extensive
Type: VLAN
Logical System: default
Routing Instance: default
Interface: demux0.1073741824
Interface type: Dynamic
Dynamic Profile Name: svlanProfile
State: Active
Session ID: 1
Stacked VLAN Id: 0x8100.1001
VLAN Id: 0x8100.1
```

```
Login Time: 2011-11-30 00:18:04 PST

Type: PPPoE
User Name: dualstackuser1@ISP1.com
IP Address: 61.1.1.1
IPv6 Prefix: 2041:1:1::/48
IPv6 User Prefix: 2061:1:1:1::/64
Logical System: default
Routing Instance: ASP-1
Interface: pp0.1073741825
Interface type: Dynamic
Dynamic Profile Name: dualStack-Profile1
MAC Address: 00:00:64:03:01:02
State: Active
Radius Accounting ID: 2
Session ID: 2
Login Time: 2011-11-30 00:18:05 PST
IPv6 Delegated Network Prefix Length: 48
IPv6 Interface Address: 2061:1:1:1::1/64
IPv6 Framed Interface Id: 1:1:2:2
IPv4 Input Filter Name: FILTER-IN-pp0.1073741825-in
IPv4 Output Filter Name: FILTER-OUT-pp0.1073741825-out
IPv6 Input Filter Name: FILTER-IN6-pp0.1073741825-in
IPv6 Output Filter Name: FILTER-OUT6-pp0.1073741825-out

Type: DHCP
IPv6 Prefix: 2041:1:1::/48
Logical System: default
Routing Instance: ASP-1
Interface: pp0.1073741825
Interface type: Static
MAC Address: 00:00:64:03:01:02
State: Active
Radius Accounting ID: jnpr :3
Session ID: 3
Underlying Session ID: 2
Login Time: 2011-11-30 00:18:35 PST
DHCP Options: len 42
00 08 00 02 0b b8 00 01 00 0a 00 03 00 01 00 00 64 03 01 02
00 06 00 02 00 19 00 19 00 0c 00 00 00 00 00 00 00 00 00 00
00 00
IPv6 Delegated Network Prefix Length: 48
```

show subscribers extensive (Effective Shaping-Rate)

```
user@host> show subscribers extensive
Type: VLAN
Logical System: default
Routing Instance: default
Interface: demux0.1073741837
Interface type: Dynamic
Interface Set: ifset-1
Underlying Interface: ae1
Dynamic Profile Name: svlan-dhcp-test
State: Active
Session ID: 1
Stacked VLAN Id: 0x8100.201
VLAN Id: 0x8100.201
Login Time: 2011-11-30 00:18:04 PST
```

```
Effective shaping-rate: 31000000k
...
```

show subscribers aci-interface-set-name detail (Subscriber Sessions Using Specified ACI Interface Set)

```
user@host> show subscribers aci-interface-set-name aci-1003-ge-1/0/0.4001 detail
Type: VLAN
Logical System: default
Routing Instance: default
Interface: ge-1/0/0.
Underlying Interface: ge-1/0/0.4001
Dynamic Profile Name: aci-vlan-set-profile
Dynamic Profile Version: 1
State: Active
Session ID: 13
Agent Circuit ID: aci-ppp-vlan-10
Login Time: 2012-03-12 10:41:56 PDT

Type: PPPoE
User Name: ppphint2
IP Address: 10.10.1.7
Logical System: default
Routing Instance: default
Interface: pp0.1073741834
Interface type: Dynamic
Interface Set: aci-1003-ge-1/0/0.4001
Interface Set Type: Dynamic
Interface Set Session ID: 13
Underlying Interface: ge-1/0/0.4001
Dynamic Profile Name: aci-vlan-pppoe-profile
Dynamic Profile Version: 1
MAC Address: 00:00:65:26:01:02
State: Active
Radius Accounting ID: 14
Session ID: 14
Agent Circuit ID: aci-ppp-vlan-10
Login Time: 2012-03-12 10:41:57 PDT
```

show subscribers agent-circuit-identifier detail (Subscriber Sessions Using Specified ACI Substring)

```
user@host> show subscribers agent-circuit-identifier aci-ppp-vlan detail
Type: VLAN
Logical System: default
Routing Instance: default
Interface: ge-1/0/0.
Underlying Interface: ge-1/0/0.4001
Dynamic Profile Name: aci-vlan-set-profile
Dynamic Profile Version: 1
State: Active
Session ID: 13
Agent Circuit ID: aci-ppp-vlan-10
Login Time: 2012-03-12 10:41:56 PDT

Type: PPPoE
User Name: ppphint2
IP Address: 10.10.1.7
Logical System: default
Routing Instance: default
Interface: pp0.1073741834
Interface type: Dynamic
Interface Set: aci-1003-ge-1/0/0.4001
```



```
Interface Set Type: Dynamic
Interface Set Session ID: 13
Underlying Interface: ge-1/0/0.4001
Dynamic Profile Name: aci-vlan-pppoe-profile
Dynamic Profile Version: 1
MAC Address: 00:00:65:26:01:02
State: Active
Radius Accounting ID: 14
Session ID: 14
Agent Circuit ID: aci-ppp-vlan-10
Login Time: 2012-03-12 10:41:57 PDT
```

show subscribers interface extensive

```
user@host> show subscribers interface demux0.1073741826 extensive
Type: VLAN
User Name: test1@test.com
Logical System: default
Routing Instance: testnet
Interface: demux0.1073741826
Interface type: Dynamic
Dynamic Profile Name: profile-vdemux-relay-23qos
MAC Address: 00:00:6e:56:01:04
State: Active
Radius Accounting ID: 12
Session ID: 12
Stacked VLAN Id: 0x8100.1500
VLAN Id: 0x8100.2902
Login Time: 2011-10-20 16:21:59 EST

Type: DHCP
User Name: test1@test.com
IP Address: 172.16.200.6
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: testnet
Interface: demux0.1073741826
Interface type: Static
MAC Address: 00:00:6e:56:01:04
State: Active
Radius Accounting ID: 21
Session ID: 21
Login Time: 2011-10-20 16:24:33 EST
Service Sessions: 2

Service Session ID: 25
Service Session Name: SUB-QOS
State: Active

Service Session ID: 26
Service Session Name: service-cb-content
State: Active
IPv4 Input Filter Name: content-cb-in-demux0.1073741826-in
IPv4 Output Filter Name: content-cb-out-demux0.1073741826-out
```

show subscribers logical-system terse

```
user@host> show subscribers logical-system test1 terse
Interface           IP Address/VLAN ID   User Name         LS:RI
demux0.1073741825     101.0.0.3            RETAILER1-CLIENT  test1:retailer1
demux0.1073741826     102.0.0.3            RETAILER2-CLIENT  test1:retailer2
```

show subscribers physical-interface count

```
user@host> show subscribers physical-interface ge-1/0/0 count
Total subscribers: 3998, Active Subscribers: 3998
```

show subscribers routing-instance inst1 count

```
user@host> show subscribers routing-instance inst1 count
Total Subscribers: 188, Active Subscribers: 183
```

show subscribers stacked-vlan-id detail

```
user@host> show subscribers stacked-vlan-id 101 detail
Type: VLAN
Interface: ge-1/2/0.1073741824
Interface type: Dynamic
Dynamic Profile Name: svlan-prof
State: Active
Stacked VLAN Id: 0x8100.101
VLAN Id: 0x8100.100
Login Time: 2009-03-27 11:57:19 PDT
```

show subscribers stacked-vlan-id vlan-id detail (Combined Output)

```
user@host> show subscribers stacked-vlan-id 101 vlan-id 100 detail
Type: VLAN
Interface: ge-1/2/0.1073741824
Interface type: Dynamic
Dynamic Profile Name: svlan-prof
State: Active
Stacked VLAN Id: 0x8100.101
VLAN Id: 0x8100.100
Login Time: 2009-03-27 11:57:19 PDT
```

show subscribers stacked-vlan-id vlan-id interface detail (Combined Output for a Specific Interface)

```
user@host> show subscribers stacked-vlan-id 101 vlan-id 100 interface ge-1/2/0.* detail
Type: VLAN
Interface: ge-1/2/0.1073741824
Interface type: Dynamic
Dynamic Profile Name: svlan-prof
State: Active
Stacked VLAN Id: 0x8100.101
VLAN Id: 0x8100.100
Login Time: 2009-03-27 11:57:19 PDT
```

show subscribers user-name detail

```
user@host> show subscribers user-name larry1 detail
Type: DHCP
User Name: larry1
IP Address: 100.0.0.37
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: ge-1/0/0.1
Interface type: Static
Dynamic Profile Name: foo
MAC Address: 00:10:94:00:00:01
State: Active
Radius Accounting ID: 1
Session ID: 1
```

```
Login Time: 2011-11-07 08:25:59 PST
DHCP Options: len 52
35 01 01 39 02 02 40 3d 07 01 00 10 94 00 00 01 33 04 00 00
00 3c 0c 15 63 6c 69 65 6e 74 5f 50 6f 72 74 20 2f 2f 32 2f
37 2d 30 2d 30 37 05 01 06 0f 21 2c
```

show subscribers vlan-id

```
user@host> show subscribers vlan-id 100
Interface           IP Address                              User Name
ge-1/0/0.1073741824
ge-1/2/0.1073741825
```

show subscribers vlan-id detail

```
user@host> show subscribers vlan-id 100 detail
Type: VLAN
Interface: ge-1/0/0.1073741824
Interface type: Dynamic
Dynamic Profile Name: vlan-prof-tpid
State: Active
VLAN Id: 100
Login Time: 2009-03-11 06:48:54 PDT

Type: VLAN
Interface: ge-1/2/0.1073741825
Interface type: Dynamic
Dynamic Profile Name: vlan-prof-tpid
State: Active
VLAN Id: 100
Login Time: 2009-03-11 06:48:54 PDT
```

show subscribers vpi vci extensive (PPPoE-over-ATM Subscriber Session)

```
user@host> show subscribers vpi 40 vci 50 extensive
Type: PPPoE
User Name: testuser
IP Address: 100.0.0.2
IP Netmask: 255.255.0.0
Logical System: default
Routing Instance: default
Interface: pp0.0
Interface type: Static
MAC Address: 00:00:65:23:01:02
State: Active
Radius Accounting ID: 2
Session ID: 2
ATM VPI: 40
ATM VCI: 50
Login Time: 2012-12-03 07:49:26 PST
IP Address Pool: pool_1
IPv6 Framed Interface Id: 200:65ff:fe23:102
```


show subscribers summary

Syntax

show subscribers summary
<all>
< detail | extensive | terse>
<count>
<physical-interface *physical-interface-name*>
<logical-system *logical-system* pic | port | routing-instance *routing-instance* | slot>

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display summary information for subscribers.

Options

all—(Optional) Display full subscriber summary.

detail | extensive | terse—(Optional) Display the specified level of output.

count—(Optional) Display the count of total subscribers and active subscribers for any specified option.

logical-system—(Optional) Display subscribers whose logical system matches the specified logical system.

physical-interface-name—(M120, M320, and MX Series routers only) (Optional) Display a count of subscribers whose physical interface matches the specified physical interface, by subscriber state, client type and LS:RI.

pic—(M120, M320, and MX Series routers only) (Optional) Display a count of subscribers by PIC number and the total number of subscribers.

port—(M120, M320, and MX Series routers only) (Optional) Display a count of subscribers by port number and the total number of subscribers.

routing-instance—(Optional) Display subscribers whose routing instance matches the specified routing instance.

slot—(M120, M320, and MX Series routers only) (Optional) Display a count of subscribers by FPC slot number and the total number of subscribers.

NOTE: Due to display limitations, logical system and routing instance output values are truncated when necessary.

Required Privilege Level

view

Related Documentation

- show subscribers on page 162

List of Sample Output

show subscribers summary on page 182

show subscribers summary all on page 182
show subscribers summary physical-interface on page 182
show subscribers summary physical-interface pic on page 183
show subscribers summary physical-interface port on page 183
show subscribers summary physical-interface slot on page 183
show subscribers summary pic on page 183
show subscribers summary pic (Aggregated Ethernet Interfaces) on page 184
show subscribers summary port on page 184
show subscribers summary slot on page 184
show subscribers summary terse on page 184

Output Fields Table 7 on page 181 lists the output fields for the **show subscribers** command. Output fields are listed in the approximate order in which they appear.

Table 7: show subscribers Output Fields

Field Name	Field Description
Subscribers by State	Number of subscribers summarized by state. The summary information includes the following: • Init—Number of subscriber currently in the initialization state. • Configured—Number of configured subscribers. • Active—Number of active subscribers. • Terminating—Number of subscribers currently terminating. • Terminated—Number of terminated subscribers. • Total—Total number of subscribers for all states.
Subscribers by Client Type	Number of subscribers summarized by client type. Client types can include DHCP, L2TP, PPP, PPPOE, STATIC-INTERFACE, and VLAN. Also displays the total number of subscribers for all client types (Total).
Subscribers by LS:RI	Number of subscribers summarized by logical system:routing instance (LS:RI) combination. Also displays the total number of subscribers for all LS:RI combinations (Total).
Interface	Interface associated with the subscriber. The router or switch displays subscribers whose interface matches or begins with the specified interface. The * character indicates a continuation of addresses for the same session. For aggregated Ethernet interfaces, the output of the **summary (pic | port | slot)** options prefixes the interface name with **ae0:**.
Count	Count of subscribers displayed for each PIC, port, or slot when those options are specified with the **summary** option. For an aggregated Ethernet configuration, the total subscriber count does not equal the sum of the individual PIC, port, or slot counts, because each subscriber can be in more than one aggregated Ethernet link.
Total Subscribers	Total number of subscribers for all physical interfaces, all PICS, all ports, or all LS:RI slots.
IP Address/VLAN ID	Subscriber IP address or VLAN ID associated with the subscriber in the form *tpid.vlan-id*
User Name	Name of subscriber.
LS:RI	Logical system and routing instance associated with the subscriber.

Sample Output

show subscribers summary

```
user@host> show subscribers summary

Subscribers by State
 Init       3
 Configured     2
 Active    183
 Terminating     2
 Terminated     1

 TOTAL         191

Subscribers by Client Type
 DHCP      107
 PPP        76
 VLAN         8

 TOTAL         191
```

show subscribers summary all

```
user@host> show subscribers summary all
Subscribers by State
 Init          3
 Configured       2
 Active      183
 Terminating       2
 Terminated       1

 TOTAL           191

Subscribers by Client Type
 DHCP       107
 PPP           76
 VLAN           8

 TOTAL          191

Subscribers by LS:RI
 default:default   1
 default:ri1     28
 default:ri2     16
 ls1:default    22
 ls1:riA     38
 ls1:riB     44
 logsysX:routinstY  42

 TOTAL     191
```

show subscribers summary physical-interface

```
user@host> show subscribers summary physical-interface ge-1/0/0
Subscribers by State
   Active: 3998
   Total: 3998

Subscribers by Client Type
   DHCP: 3998
```



```
   Total: 3998

Subscribers by LS:RI
   default:default: 3998
   Total: 3998
```

show subscribers summary physical-interface pic

```
user@host> show subscribers summary physical-interface ge-0/2/0 pic
Subscribers by State
   Active: 4825
   Total: 4825

Subscribers by Client Type
   DHCP: 4825
   Total: 4825

Subscribers by LS:RI
   default:default: 4825
   Total: 4825
```

show subscribers summary physical-interface port

```
user@host> show subscribers summary physical-interface ge-0/3/0 port
Subscribers by State
   Active: 4825
   Total: 4825

Subscribers by Client Type
   DHCP: 4825
   Total: 4825

Subscribers by LS:RI
   default:default: 4825
   Total: 4825
```

show subscribers summary physical-interface slot

```
user@host> show subscribers summary physical-interface ge-2/0/0 slot
Subscribers by State
   Active: 4825
   Total: 4825

Subscribers by Client Type
   DHCP: 4825
   Total: 4825

Subscribers by LS:RI
   default:default: 4825
   Total: 4825
```

show subscribers summary pic

```
user@host> show subscribers summary pic
Interface           Count
ge-1/0              1000
ge-1/3              1000

Total Subscribers: 2000
```

show subscribers summary pic (Aggregated Ethernet Interfaces)

```
user@host> show subscribers summary pic
Interface           Count
ae0: ge-1/0         801
ae0: ge-1/3         801

Total Subscribers: 801
```

show subscribers summary port

```
user@host> show subscribers summary port
Interface           Count
ge-1                2000

Total Subscribers: 2000
```

show subscribers summary slot

```
user@host> show subscribers summary slot
Interface           Count
ge-1                2000

Total Subscribers: 2000
```

show subscribers summary terse

```
user@host> show subscribers summary terse
Interface           IP Address/VLAN ID   User Name         LS:RI
ge-1/3/0.1073741824   100                                    default:default
demux0.1073741824     100.0.0.10           WHOLESALER-CLIENT default:default
demux0.1073741825     101.0.0.3            RETAILER1-CLIENT  test1:retailer1
demux0.1073741826     102.0.0.3            RETAILER2-CLIENT  test1:retailer2
```


PART 4

Troubleshooting

CHAPTER 14

Monitoring and Logs

Collecting Subscriber Access Logs Before Contacting Juniper Technical Support

Problem When you experience a subscriber access problem in your network, we recommend that you collect certain logs before you contact Juniper Technical Support. This topic shows you the most useful logs for a variety of network implementations. In addition to the relevant log information, you must also collect standard troubleshooting information and send it to Juniper Technical Support in your request for assistance.

Solution To collect standard troubleshooting information:

- Redirect the command output to a file.

  user@host> **request support information | save rsi-1**

To configure logging to assist Juniper Technical Support:

1. Review the following blocks of statements to determine which apply to your configuration.

   [edit]
   set system syslog archive size 100m files 25
   set system auto-configuration traceoptions file *filename*
   set system auto-configuration traceoptions file filename size 100m files 25
   set protocols ppp-service traceoptions file *filename* size 100m files 25
   set protocols ppp-service traceoptions level all
   set protocols ppp-service traceoptions flag all
   set protocols ppp traceoptions file *filename* size 100m files 25
   set protocols ppp traceoptions level all
   set protocols ppp traceoptions flag all
   set protocols ppp monitor-session all
   set interfaces pp0 traceoptions flag all
   set demux traceoptions file *filename* size 100m files 25
   set demux traceoptions level all
   set demux traceoptions flag all
   set system processes dhcp-service traceoptions file *filename*
   set system processes dhcp-service traceoptions file size 100m
   set system processes dhcp-service traceoptions file files 25
   set system processes dhcp-service traceoptions flag all
   set class-of-service traceoptions file *filename*
   set class-of-service traceoptions file size 100m
   set class-of-service traceoptions flag all
   set class-of-service traceoptions file files 25
   set routing-options traceoptions file *filename*
   set routing-options traceoptions file size 100m
   set routing-options traceoptions flag all
   set routing-options traceoptions file files 25
   set interfaces traceoptions file *filename*
   set interfaces traceoptions file size 100m
   set interfaces traceoptions flag all
   set interfaces traceoptions file files 25
   set system processes general-authentication-service traceoptions file *filename*
   set system processes general-authentication-service traceoptions file size 100m
   set system processes general-authentication-service traceoptions flag all
   set system processes general-authentication-service traceoptions file files 25
2. Copy the relevant statements into a text file and modify the log filenames as you want.
3. Copy the statements from the text file and paste them into the CLI on your router to configure logging.
4. Commit the logging configuration to begin collecting information.

NOTE: The maximum file size for DHCP local server and DHCP relay log files is 1 GB. The maximum number of log files for DHCP local server and DHCP relay is 1000.

BEST PRACTICE: Enable these logs only to collect information when troubleshooting specific problems. Enabling these logs during normal operations can result in reduced system performance.

Related Documentation

- *Compressing Troubleshooting Logs from /var/logs to Send to Juniper Technical Support*

PART 5

Index

Index

M

N

P