Engineering
Simplicity

Junos® OS

Broadband Subscriber VLANs and Interfaces User Guide

Published
2025-12-19

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

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

Table of Contents

About This Guide

Use this guide to learn how to configure the logical portion of subscriber management networks to provision services using virtual local area networks (VLANs) with DHCP, PPPoE, and MLPPP interfaces.

1

PART

Configuring Dynamic VLANs for Subscriber Access Networks

CHAPTER 1

Dynamic VLAN Overview

IN THIS CHAPTER

Subscriber Management VLAN Architecture Overview

IN THIS SECTION

The subscriber management logical network architecture is as important as the physical network architecture. You configure the logical portion of the subscriber management network using virtual local area networks (VLANs).

Customer VLANs

Customer VLANs (C-VLANs) provide one-to-one (1:1) subscriber-to-service connectivity: One VLAN carries all traffic to each subscriber on the network. Having a single VLAN per subscriber simplifies operations by providing a 1:1 mapping of technology (VLANs) to subscribers. You can also understand what applications any subscriber is using at any given time. Because you use only one VLAN to carry traffic to each subscriber, this approach is not affected when adding new services. However, using a pure C-VLAN model consumes more bandwidth because a single television channel being viewed by multiple subscribers is carried across the network several times—once on each C-VLAN. This approach requires a more scalable, robust edge router that can support several thousand VLANs.

Configurations that use C-VLANs uniquely identify subscribers by using the VLAN ID and stacked VLAN (S-VLAN) ID. Subscriber packets received from the access node that are either single-tagged with a VLAN ID or double-tagged with both an S-VLAN ID and a VLAN ID are examples of C-VLAN configurations because they provide a one-to-one correspondence between an individual subscriber and the VLAN encapsulation.

In the C-VLAN architecture, each customer premises equipment (CPE) or subscriber network has its own dedicated Layer 2 path to the router. Each subscriber network is separated by a customer VLAN (C-VLAN) that is dedicated to a particular customer. The services for each customer are transmitted from the router to the access node by means of that customer's C-VLAN.

The ability to uniquely identify subscribers by means of VLAN encapsulation facilitates delivery of services such as authentication, authorization, and accounting (AAA); *class of service* (CoS); and filters (policers) to subscribers in a C-VLAN configuration.

We recommend using C-VLANs for data and voice traffic to simplify configuration and management when expanding services. However, some MSANs are limited to the number of VLANs they can support, limiting the ability to use C-VLANs.

Service VLANs

Service VLANs (S-VLANs) provide many-to-one (N:1) subscriber-to-service connectivity: The service VLAN carries a service (for example, data, video, or voice) to all subscribers instead of having different services share a VLAN. Adding a new service requires adding a new VLAN and allocating bandwidth to the new service. The service VLAN model enables different groups that are using the broadband network (for example, external application providers) to manage a service. One limitation of service VLANs is the absence of any logical isolation between user sessions at the VLAN level. This lack of isolation requires that the multiservice access node (MSAN) and broadband network gateway (BNG) provide the necessary security filtering.

Service VLANs enable service providers to route different services to different routers to functionally separate network services and reduce network complexity.

Typically, you would use S-VLANs for video and IPTV traffic.

Hybrid VLANs

Hybrid C-VLAN—The hybrid VLAN combines the best of both previous VLANs by using one VLAN per subscriber to carry unicast traffic and one shared multicast VLAN (M-VLAN) for carrying broadcast (multicast) television traffic. You can use both the *pure* and *hybrid* C-VLAN models in different portions of the network, depending upon available bandwidth and MSAN capabilities.

The term *C-VLAN*, when used casually, often refers to a *hybrid* C-VLAN implementation.

Broadband Subscriber Management VLANs Across an MSAN

You configure VLANs to operate between the MSAN and the edge router (broadband services router or video services router). However, the MSAN might modify VLAN identifiers before forwarding information to the subscriber in the following ways:

- The VLAN identifiers can be carried within the ATM VCs or they can be removed. The value of keeping the VLAN header is that it carries the IEEE 802.1p Ethernet priority bits. These priority bits can be added to upstream traffic by the residential gateway, allowing the DSLAM to easily identify and prioritize more important traffic (for example, control and VoIP traffic). Typically, a VLAN identifier of zero (0) is used for this purpose.
- In a C-VLAN model, the MSAN might modify the VLAN identifier so that the same VLAN is sent to each subscriber. This enables the use of the same digital subscriber line (DSL) modem and residential gateway configuration for all subscribers without the need to define a different VLAN for each device.

Not all MSANs support these options. Most MSANs can support the service VLAN model.

Customer VLANs and Ethernet Aggregation

The 12-bit VLAN identifier (VLAN ID) can support up to 4095 subscribers. When using an aggregation switch with a C-VLAN topology, and fewer than 4095 subscribers are connected to a single edge router port, the aggregation switch can transparently pass all VLANs. However, if the VLAN can exceed 4095 subscribers per broadband services router port, you must use VLAN stacking (IEEE 802.1ad, also known as Q-in-Q). VLAN stacking includes two VLAN tags—an outer tag to identify the destination MSAN and an inner tag to identify the subscriber. For downstream traffic (that is, from the broadband services router or Ethernet switch to the MSAN), the outer tag determines which port to forward traffic. The forwarding device then uses the VLAN pop function on this tag before forwarding the traffic with a single tag. The reverse process occurs for upstream traffic.

VLAN stacking is not necessary for S-VLANs or M-VLANs. However, for the hybrid (C-VLAN and M-VLAN) model, the Ethernet switch or services router must be able to pop or push tags onto C-VLAN traffic while not modifying M-VLAN packets.

RELATED DOCUMENTATION

Static Subscriber Interfaces and VLAN Overview | 7

Dynamic 802.1Q VLAN Overview

IN THIS SECTION

You can identify VLANs statically or dynamically. You can also configure a mix of static and dynamic VLANs on the same underlying interface.

For Ethernet, Fast Ethernet, Tri-Rate Ethernet copper, Gigabit Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet interfaces supporting VPLS, Junos OS supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces. Many hosts can be connected to the same Gigabit Ethernet switch, but they cannot be in the same routing or bridging domain.

To identify VLANs statically, you can reference a static VLAN interface in a dynamic profile. To identify subscribers dynamically, you use a variable to specify an 802.1Q VLAN that is dynamically created when a subscriber accesses the network.

Dynamic VLAN Configuration

You can configure the router to dynamically create VLANs when a client accesses an interface and requests a VLAN ID that does not yet exist. When a client accesses a particular interface, the router instantiates a VLAN dynamic profile that you have associated with the interface. Using the settings in the dynamic profile, the router extracts information about the client from the incoming packet (for example, the interface and unit values), saves this information in the routing table, and creates a VLAN or stacked VLAN ID for the client from a range of VLAN IDs that you configure for the interface.

Dynamic VLAN configuration supports the creation of IPv4 (inet), DHCPv4, IPv6 (inet6), and DHCPv6 VLANs.

Dynamic Mixed VLAN Ranges

Dynamic VLAN and dynamic stacked VLAN configuration supports mixed (or flexible) VLAN ranges. When you configure dynamic mixed VLAN ranges, you must create separate dynamic profiles for VLANs and stacked VLANs. Table 1 on page 6 lists all valid combinations for the maximum number of dynamic profiles and VLAN and stacked VLAN ranges on a single underlying interface.

Table 1: Maximum Dynamic Profiles and Ranges for Dynamic Mixed VLAN Configurations

VLANs		Stacked VLANs	
Maximum Number of Dynamic Profiles	Maximum Number of VLAN Ranges Per Profile	Maximum Number of Dynamic Profiles	Maximum Number of Stacked VLAN Ranges Per Profile
1	128	1	128
16	32	16	32
1	128	16	32
16	32	1	128

Table 1 on page 6 shows the valid maximums for the following dynamic mixed VLAN range configuration scenarios, in this order:

- Configurations that require up to 128 VLAN ranges and up to 128 stacked VLAN ranges on a single underlying interface. You must create one VLAN dynamic profile and one stacked VLAN dynamic profile, each with a maximum of 128 ranges per profile.
- Configurations that require up to 32 VLAN ranges and up to 32 stacked VLAN ranges on a single underlying interface. You can configure up to 16 VLAN dynamic profiles and up to 16 stacked VLAN dynamic profiles, each with a maximum of 32 ranges per profile.
- Configurations that consist of one VLAN dynamic profile with a maximum of 128 ranges, and up to 16 stacked VLAN dynamic profiles with 32 ranges each.
- Configurations that consist of up to 16 VLAN dynamic profiles with 32 ranges each, and one stacked VLAN dynamic profile with a maximum of 128 ranges.

The following guidelines apply to the limits in Table 1 on page 6 when you configure VLAN ranges and S-VLAN ranges for use with dynamic profiles:

- These limits apply to both single-tagged and double-tagged dynamic VLAN ranges.
- These limits apply only to MX Series routers with MPCs. For MX Series routers with Enhanced Queuing IP Services DPCs (DPCE-R-Q model numbers) or Enhanced Queuing Ethernet Services DPCs (DPCE-X-Q model numbers), the maximum number of VLAN ranges for a dynamic profile on an underlying interface remains unchanged at 32 VLAN ranges and 32 S-VLAN ranges.
- These limits have no effect on the maximum number of VLAN IDs on a given underlying interface. The valid range of ID values for a dynamic VLAN range or dynamic S-VLAN range remains unchanged at 1 through 4094.

RELATED DOCUMENTATION

Configuring Interfaces to Support Both Single and Stacked VLANs | 23

Static Subscriber Interfaces and VLAN Overview

This topic describes the topology for configuring subscriber interfaces over static VLAN interfaces.

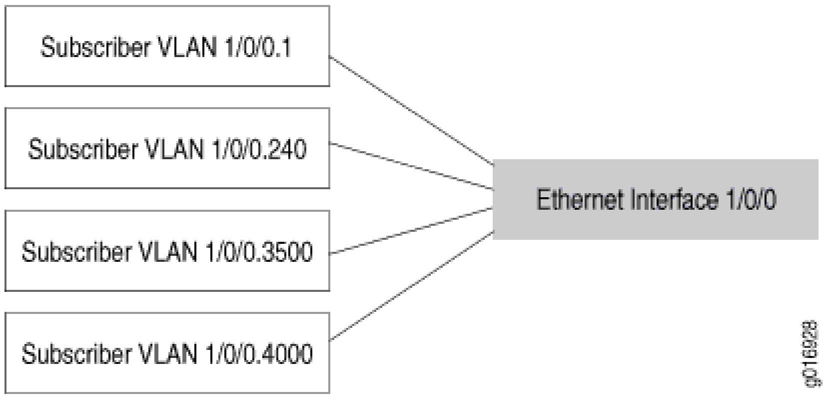
In a dynamic profile, you can configure VLAN subscriber interfaces over the following statically created *logical interface* types:

- GE—Gigabit Ethernet
- XE—10-Gigabit Ethernet
- AE—Aggregated Ethernet

We recommend that you configure each subscriber on a statically created VLAN.

Figure 1 on page 8 shows an example of subscriber interfaces on an individual VLAN.

Figure 1: VLAN Subscriber Interfaces



You can further separate VLANs on subscriber interfaces by configuring a VLAN interface as the underlying interface for a set of IP demux interfaces.

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

Pseudowire Termination: Explicit Notifications for Pseudowire Down Status

As the demand for MPLS-based Layer 2 services grows, new challenges arise for service providers to be able to interoperate Layer 2 with Layer 3 and give their customers value-added services. MPLS in the access networks is already used by applications like mobile or DSL backhaul to achieve a more cost-efficient solution, better service reliability, and quality of service. Most of the traditional access network infrastructure is built over TDM circuits such as DS3 for higher speeds, ATM, or Frame Relay as access trails in a Layer 3 service. For higher bandwidth requirements and more flexibility, service providers use Ethernet as access technology for a wide range of network services. Although Ethernet provides a convenient link topology for access networks, it is not well suited for Layer 2 switching and for aggregating traffic from the access network to the core. MPLS is already used in the core and now its presence in the access network enables use of a single technology across the network. When MPLS is deployed in the access network, Ethernet is used as a link-layer encapsulation technology only, and MPLS switches perform traffic forwarding and provide other Layer 2 services. There is an increase in demand for using pseudowires as access circuits in the service delivery points in the network. These pseudowires terminate on a service node on which the service provider applies Layer 3 or Layer 2 services to the customer data.

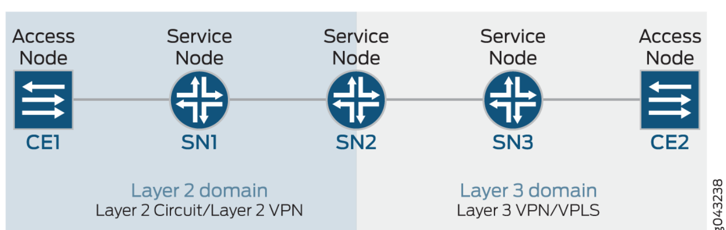
The following is a generic topology for understanding termination for pseudowire into a Layer 2 or Layer 3 instance and the notifications for both cases.

The following terminologies are used for the network elements:

- Access node (AN): An access node is typically a customer edge device that processes the packets entering or exiting the network at Layer 2. This includes devices such as DSLAMs and MSANs.
- Transport node(TN): A transport node acts like a P router as it does not have any customer or service state. It is either used for connecting the access node to the service node or to two service nodes.
- Service node (SN): A service node is a PE router that applies services to the customer packets. It includes Layer 2 PE, Layer 3 PE, peering routers, video servers, base station controllers, and media gateways.

The following example shows a linear L2-L3 interconnection set up with the absence of pseudowire redundancy. Here, the access circuit pseudowire is configured between the access PE (SN1) and service node (SN2), which defines the boundary of the L2 domain. The Layer 3 VPN is configured between SN2 and SN3, which constitute the L3 domain. Layer 2 circuit pseudowire terminates in the VRF of the device interconnecting the L2-L3 domains (SN2); that is, the service node performs stitching between the Layer 2 circuit and the Layer 3 VPN.

Figure 2: Pseudowire Termination



RELATED DOCUMENTATION

Configuring an Access Pseudowire That Terminates into VRF on the Service Node | 10

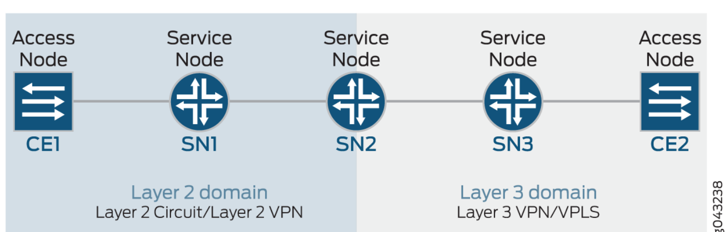
Configuring an Access Pseudowire That Terminates into a VPLS Routing Instance | 13

Configuring an Access Pseudowire That Terminates into VRF on the Service Node

Each VPN has its own VPN-specific routing table per VPN site. When an ingress PE router (SN2) receives routes advertised from a directly connected access node (CE2), it checks the received route against the VRF export policy for that VPN. If it matches, the route is converted to VPN-IPv4 format; that is, the route distinguisher is added to the route. This VPN-IPv4 route is advertised to the remote PE routers. It also attaches a route target to each route learned from the directly connected sites, which is based on the value of the configured export target policy of the VRF tables. When an egress PE router receives this route, it checks it against the import policy between the PE routers. If accepted, the route is placed into its `bgp.l3vpn.0` table. At the same time, the router checks the route against the VRF import policy for the VPN. If it matches, the route distinguisher is removed from the route, and the route is placed into the VRF table in IPv4 format.

On SN2 and SN1, routes are installed in the VRF based on the import and export VRF policies. OSPF and direct routes from CE2 are installed in the VRF of SN2, which is then converted into IPv4-VPN routes. The routes to be learned over the CE-PE link is defined under protocols in the routing instance. Now, from the other end, the access pseudowire terminates in the VRF of the SN1 device, and the static routing is configured between the access node (CE1) and the service node(SN1). Traffic at this point is handled at the IP level, before it enters the Layer 3 domain. The translation from IP route to IPv4-VPN route happens at SN2.

Figure 3: Pseudowire Termination



1. To configure the logical tunnel interfaces or the lt-ifls.

```
[edit interfaces]
lt-0/0/10 {
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id number;
        peer-unit 1;
```

```
    }
    unit 1 {
        encapsulation vlan;
        vlan-id number;
        peer-unit 0;
        family inet {
            address IPv4 address;
        }
    }
}
```

2. To configure appropriate import and export policies.

   Each VPN has its own VPN-specific routing table per VPN site. When an ingress PE router (CE2) receives routes advertised from a directly connected access node, it checks the received route against the VRF export policy for that VPN. If it matches, the route is converted to VPN-IPv4 format; that is, the route distinguisher is added to the route.

```
[edit policy-options]
policy-statement policy-name {
    term 1 {
        from protocol [ direct ospf ];
        then {
            community add l3vpn;
            accept;
        }
    }
}
```

   When an egress router receives this route, it checks it against the import policy between the CE routers. If it is accepted, then the route is placed into its `bgp.l3vpn.0` table. At the same time, the router checks the route against the VRF import policy for the VPN.

```
[edit policy-options]
policy-statement policy-name {
    term 1 {
        from community l3vpn;
        then accept;
    }
}
```

3. To access the pseudowire configuration on SN1.

```
[edit protocols]
l2circuit {
    neighbor address {
        interface lt-0/0/10.0 {
            virtual-circuit-id number;
        }
    }
}
```

4. To configure the Layer 3 VPN routing instance.

   In Layer 2 domains where service node SN1 interconnects the L2 to L3 domain, you need to activate the `vrf-table-label` feature to be able to advertise the direct-subnet prefix that corresponds to the lt-ifl toward the Layer 3 domain.

```
[edit routing-instances]
l3vpn routing instance {
    instance-type vrf;
    interface lt-0/0/10.1;
    route-distinguisher 100:2;
    vrf-import l3vpn-import;
    vrf-export l3vpn-export;
    vrf-table-label;
    protocols {
        ospf {
            export ospf_export;
            area 0.0.0.0 {
                interface all {
                    priority 0;
                }
            }
        }
    }
}
```

Use the following operational mode commands to verify termination of an access pseudowire into VRF:

- `show l2circuit connections`
- `show route table l3vpn_1.inet.0`

RELATED DOCUMENTATION

Pseudowire Termination: Explicit Notifications for Pseudowire Down Status | 8

Configuring an Access Pseudowire That Terminates into a VPLS Routing Instance | 13

Configuring an Access Pseudowire That Terminates into a VPLS Routing Instance

Terminating the access pseudowire into a VPLS instance is supported for both LDP-VPLS and BGP-VPLS.

To configure an access pseudowire that terminates into VPLS on the service node using LT-IFLS and mesh-groups:

1. Configure the logical tunnel interfaces or the lt-ifls.

   Logical tunnel interface pairs are used for stitching Layer 2 network elements to VPLS when an access pseudowire terminates into a VPLS routing instance.

```
[edit interfaces]
interface name {
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id number;
        peer-unit 1;
    }
    unit 1 {
        encapsulation vlan-vpls;
        vlan-id number;
        peer-unit 0;
        family vpls;
    }
}
```

2. Configure the VPLS routing instance.

   To terminate the access pseudowire into a VPLS routing instance, use mesh groups as follows:

```
[edit routing-instances]
routing-instance name {
    instance-type vpls;
    interface interface name;
```

```
    route-distinguisher 192.0.2.255:1;
    vrf-target target:64577:1;
    protocols {
        site vpls {
            site-identifier 4;
            interface interface name;
        }
        mesh-group pe-mid {
            vpls-id number;
            local-switching;
            neighbor 192.0.2.1;
        }
    }
}
```

In LDP-VPLS and BGP-VPLS, the Layer 2 circuit only needs to be configured on the access PE (SN1) with a virtual circuit ID, and the corresponding VPLS ID is configured on the service node for terminating the pseudowire. Local switching can be used on the service node to switch the traffic from multiple pseudowires into the desired VPLS routing instance.

Use the `show vpls connections` operational mode command to verify termination of an access pseudowire into a VPLS routing instance.

RELATED DOCUMENTATION

Pseudowire Termination: Explicit Notifications for Pseudowire Down Status | **8**

Configuring an Access Pseudowire That Terminates into VRF on the Service Node | **10**

CHAPTER 2

Configuring Dynamic Profiles and Interfaces Used to Create Dynamic VLANs

IN THIS CHAPTER

Configuring a Dynamic Profile Used to Create Single-Tag VLANs

You can configure a dynamic profile for creating single-tagged VLANs.

To configure a dynamic VLAN profile:

1. Create a dynamic profile.

```
user@host# set dynamic-profile VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"
```

2. (Optional) To support dynamic demux interfaces, enable them for IPv4 or IPv6.

- For IPv4 demux interfaces:

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set demux-source inet
```

- For IPv6 demux interfaces:

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set demux-source inet6
```

3. (Optional) To configure the router to respond to any ARP request, specify the `proxy-arp (Dynamic Profiles)` statement.

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set proxy-arp (Dynamic Profiles)
```

4. Specify that you want to use dynamic VLAN IDs in the dynamic profile. You can configure the dynamic profile to create a single-tag VLAN using only standard tag protocol identifier (TPID) values (0x8100) or to create a VLAN using any TPID value.

- To configure the dynamic profile to create single-tag VLANs that accept only standard TPID values (a TPID value of 0x8100):

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set vlan-id $junos-vlan-id
```

When the dynamic profile is instantiated, the variable is dynamically replaced with a VLAN ID within the VLAN range specified at the `[interfaces]` hierarchy level.

- To configure the dynamic profile to create single-tag VLANs that accept any TPID value:

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-interface-unit"]
user@host# set vlan-tags outer $junos-vlan-id
```

The variable is dynamically replaced with both the TPID value and a VLAN ID within the VLAN range specified at the `[interfaces]` hierarchy level.

5. Define the unit family type.

   a. For IPv4 interfaces:

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set family inet
```

   b. For IPv6 interfaces:

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set family inet6
```

6. (Optional) Enable IP and MAC address validation for dynamic demux interfaces in a dynamic profile.

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" family inet]
user@host# set mac-validate loose
```

7. Specify the unnumbered address and preferred source address.

```
[edit dynamic-profiles VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" family inet]
user@host# set unnumbered-address lo.0 preferred-source-address 192.0.2.16
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure a dynamic profile for creating single-tagged VLANs.

RELATED DOCUMENTATION

Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs | 18

Dynamic 802.1Q VLAN Overview | 5
Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57

Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs

Starting in Junos OS Release 14.1, you configure an interface to use a dynamic profile when the dynamic VLANs are created. The dynamic profile uses the VLAN ranges configured for the interface.

To configure the interface:

1. Specify the interface over which you want to create dynamic VLANs.

```
user@host# edit interfaces ge-0/0/0
```

2. Specify the VLAN range configuration.

```
[edit interfaces ge-0/0/0]
user@host# edit auto-configure vlan-ranges
```

3. Specify the dynamic profile used to create VLANs. To create the dynamic profile, see Configuring a Dynamic Profile Used to Create Single-Tag VLANs.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# edit dynamic-profile VLAN-PROF1
```

4. Specify the VLAN Ethernet packet type the VLAN dynamic profile accepts.

   `inet` and `dhcp-v4` for IPv4 packets, `inet6` and `dhcp-v6` for IPv6 packets, and `pppoe` for PPP packets are supported.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges dynamic-profile VLAN-PROF1]
user@host# set accept inet
```

5. Specify the VLAN ranges that you want the dynamic profile to use. The following example specifies a lower VLAN ID limit of 3000 and any upper VLAN ID limit (a range from 1 through 4094).

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges dynamic-profile VLAN-PROF1]
user@host# set ranges 3000-any
```

NOTE: You can configure multiple VLAN range groups (up to 32 total) on the same physical interface that use different VLAN dynamic profiles.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you configure an interface to use a dynamic profile when the dynamic VLANs are created. The dynamic profile uses the VLAN ranges configured for the interface.

RELATED DOCUMENTATION

Configuring a Dynamic Profile Used to Create Single-Tag VLANs | 15
Dynamic 802.1Q VLAN Overview | 5

Configuring a Dynamic Profile Used to Create Stacked VLANs

Starting in Junos OS Release 14.1, you can configure a dynamic profile for creating stacked 802.1Q VLANs.

To configure a dynamic VLAN profile:

1. Create a dynamic profile.

```
user@host# set dynamic-profile STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit
"$junos-interface-unit"
```

2. (Optional) To support dynamic demux interfaces, enable them for IPv4 or IPv6.

- For IPv4 demux interfaces:

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit
"$junos-interface-unit"]
user@host# set demux-source inet
```

- For IPv6 demux interfaces:

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit
"$junos-interface-unit"]
user@host# set demux-source inet6
```

3. (Optional) To configure the router to respond to any ARP request, specify the `proxy-arp` statement.

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set proxy-arp
```

4. Specify the outer VLAN ID variable.

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set vlan-tags outer $junos-stacked-vlan-id
```

   The variable is dynamically replaced with an outer VLAN ID within the VLAN range specified at the `[interfaces]` hierarchy level.

5. Specify the inner VLAN ID variable.

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit"]
user@host# set vlan-tags inner $junos-vlan-id
```

   The variable is dynamically replaced with an inner VLAN ID within the VLAN range specified at the `[interfaces]` hierarchy level.

6. Define the unit family type.

a. For IPv4 interfaces:

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit
"$junos-interface-unit"]
user@host# set family inet
```

b. For IPv6 interfaces:

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit
"$junos-interface-unit"]
user@host# set family inet6
```

7. (Optional) Enable IP and MAC address validation for dynamic demux interfaces in a dynamic profile.

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" family inet]
user@host# set mac-validate loose
```

8. Specify the unnumbered address and preferred source address.

```
[edit dynamic-profiles STACKED-VLAN-PROF1 interfaces "$junos-interface-ifd-name" unit "$junos-
interface-unit" family inet]
user@host# set unnumbered-address lo.0 preferred-source-address 192.0.2.16
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure a dynamic profile for creating stacked 802.1Q VLANs.

RELATED DOCUMENTATION

Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs | 22

Configuring a Basic Dynamic Profile

Dynamic 802.1Q VLAN Overview | 5

Dynamic Variables Overview

Junos OS Predefined Variables

Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs

Starting in Junos OS Release 14.1, you configure an interface to use a dynamic profile when the dynamic VLANs are created. The dynamic profile uses the VLAN ranges configured for the interface.

To configure the interface:

1. Specify the interface over which you want to create dynamic VLANs.

```
user@host# edit interfaces ge-0/0/0
```

2. Specify that this interface is for use with stacked VLAN ranges.

```
[edit interfaces ge-0/0/0]
user@host# set stacked-vlan-tagging
```

3. Specify the VLAN range configuration.

```
[edit interfaces ge-0/0/0]
user@host# edit auto-configure stacked-vlan-ranges
```

4. Specify the dynamic profile used to create VLANs. To create the dynamic profile, see Configuring a Dynamic Profile Used to Create Stacked VLANs.

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges]
user@host# edit dynamic-profile STACKED-VLAN-PROF1
```

5. Specify the VLAN Ethernet packet type the VLAN dynamic profile accepts.

inet and dhcp-v4 for IPv4 packets, inet6 and dhcp-v6 for IPv6 packets, and pppoe for PPP packets are supported.

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges dynamic-profile STACKED-VLAN-PROF1]
user@host# set accept inet
```

6. Specify the outer and inner stacked VLAN ranges that you want the dynamic profile to use. The following example specifies an outer stacked VLAN ID range from 2000 through 4000 and an inner stacked VLAN ID range of any (enabling a range from 1 through 4094 for the inner stacked VLAN ID).

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges dynamic-profile STACKED-VLAN-PROF1]
user@host# set ranges 2000-4000,any
```

NOTE: You can configure multiple dynamic profile associations (up to 16) with different VLAN range groups on each physical interface.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you configure an interface to use a dynamic profile when the dynamic VLANs are created. The dynamic profile uses the VLAN ranges configured for the interface.

RELATED DOCUMENTATION

Configuring a Dynamic Profile Used to Create Stacked VLANs | 19
Dynamic 802.1Q VLAN Overview | 5

Configuring Interfaces to Support Both Single and Stacked VLANs

You can configure VLANs to support simultaneous transmission of 802.1Q VLAN single-tag and stacked frames on logical interfaces on the same Ethernet port, and on pseudowire logical interfaces.

Junos VLAN IDs for single-tag VLANs are equivalent to the outer tags used for stacked (dual-tag) VLANs. When configuring mixed (flexible) VLANs, any overlap on single-tag VLAN IDs and stacked VLAN outer tag values is supported only for dynamic VLANs on MPC line cards. When configuring mixed (flexible) VLANS on DPCE line cards, overlapping single-tag VLAN IDs and stacked VLAN outer tag values is not supported. This means that a dynamically created single-tagged VLAN interface prevents any overlapping stacked VLAN interfaces from being created or a dynamically created stacked VLAN interface prevents any overlapping single-tagged VLAN interfaces from being created.

NOTE: For information about the maximum number of dynamic profiles, VLAN ranges, and stacked VLAN ranges for dynamic mixed VLAN configurations, see "Dynamic 802.1Q VLAN Overview" on page 5.

To configure both VLAN and stacked VLAN ranges:

1. Specify the interface over which you want to create dynamic VLANs.

```
user@host# edit interfaces ge-0/0/0
```

2. Indicate that this interface is for use with both VLAN and stacked VLAN ranges.

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

3. Define interface automatic configuration values.

```
[edit interfaces ge-0/0/0]
user@host# edit auto-configure
```

4. Specify that you want to modify VLAN ranges.

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit vlan-ranges
```

5. Specify the VLAN dynamic profile for which you want to configure VLAN ranges. To create the dynamic profile, see Configuring a Dynamic Profile Used to Create Single-Tag VLANs.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# edit dynamic-profile VLAN-PROF1
```

6. Specify the VLAN ranges that you want the dynamic profile to use. The following example specifies a lower VLAN ID limit of 2000 and an upper VLAN ID limit of 3000.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges dynamic-profile VLAN-PROF1]
user@host# set ranges 2000-3000
```

NOTE: You can configure multiple dynamic profile associations (up to 32) with different VLAN range groups on each physical interface.

7. Specify that you want to modify stacked VLAN ranges.

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit stacked-vlan-ranges
```

8. Specify the stacked VLAN dynamic profile for which you want to configure VLAN ranges. To create the stacked VLAN dynamic profile, see Configuring a Dynamic Profile Used to Create Stacked VLANs.

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges]
user@host# edit dynamic-profile STACKED-VLAN-PROF1
```

9. Specify the outer and inner stacked VLAN ranges that you want the dynamic profile to use. The following example specifies an outer stacked VLAN ID range from 3001 through 4000 (to avoid overlapping VLAN IDs with single-tag VLANs) and an inner stacked VLAN ID range of `any` (enabling a range from 1 through 4094 for the inner stacked VLAN ID).

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges dynamic-profile STACKED-VLAN-
PROF1]
user@host# set ranges 3001-4000,any
```

NOTE: You can configure multiple dynamic profile associations (up to 32) with different VLAN range groups on each physical interface.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, you can configure VLANs to support simultaneous transmission of 802.1Q VLAN single-tag and stacked frames on logical interfaces on the same Ethernet port, and on pseudowire logical interfaces.

RELATED DOCUMENTATION

Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs | 18
Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs | 22
Dynamic 802.1Q VLAN Overview | 5

Overriding the Dynamic Profile Used for an Individual VLAN

You can override dynamic profile assignment to individual VLANs that are already part of a previously defined VLAN range. This functionality provides a type of exception to an assigned VLAN range. It enables you to configure individual VLAN IDs to use a different dynamic profile from the one assigned to the VLAN range that includes the individual VLAN ID.

To configure dynamic profile override for a specific VLAN:

1. Access the interface on which you want to create a dynamic profile override.

```
user@host# edit interfaces ge-0/0/0
```

2. Access the interface automatic configuration hierarchy.

```
[edit interfaces ge-0/0/0]
user@host# edit auto-configure
```

3. Access either the single-tagged or dual-tagged (stacked) VLAN ranges that you want to modify.

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit vlan-ranges
```

or

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit stacked-vlan-ranges
```

4. Define the `override` statement along with the VLAN tag that you want to override and the dynamic profile that you want to use when overriding the specified VLAN tag.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# set override tag 20 dynamic-profile NewProfile
```

or

```
[edit interfaces ge-0/0/0 auto-configure stacked-vlan-ranges]
user@host# set override tag 20 dynamic-profile NewProfile
```

Configuring a VLAN Dynamic Profile That Associates VLANs with Separate Routing Instances

You can configure a VLAN dynamic profile that dynamically creates underlying VLAN interfaces and associates these interfaces with statically created routing instances. The VLAN interface is created for a specific routing instance as defined by VSA 26–1 (Virtual-Router) on the AAA server (for example, RADIUS server).

To configure a dynamic VLAN profile to use routing instances when creating VLANs, add the routing instance configuration to your dynamic profile:

1. Access the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles VLAN_PROFILE_RI
```

2. Specify that you want to dynamically associate the profile with routing instances.

```
[edit dynamic-profiles VLAN_PROFILE_RI]
user@host# edit routing-instances $junos-routing-instance
```

3. Define the routing instance `interface` statement with the internal `$junos-interface-name` variable used by the router to match the interface name of the receiving interface.

```
[edit dynamic-profiles VLAN_PROFILE_RI routing-instances “$junos-routing-instance”]
user@host# set interface $junos-interface-name
```

4. Define the dynamic profile `interfaces` statement with the internal `$junos-interface-ifd-name` variable.

```
[edit dynamic-profiles VLAN_PROFILE_RI]
user@host# edit interfaces $junos-interface-ifd-name
```

5. Define the `unit` statement with the internal `$junos-interface-unit` variable used by the router to generate a unit value for the interface.

```
[edit dynamic-profiles VLAN_PROFILE_RI interfaces "$junos-interface-ifd-name"]
user@host# edit unit $junos-interface-unit
```

RELATED DOCUMENTATION

Configuring a Basic Dynamic Profile
Dynamic 802.1Q VLAN Overview | 5
Dynamic Variables Overview
Junos OS Predefined Variables
Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames
Configuring Dynamic Authentication for VLAN Interfaces | 33

Automatically Removing VLANs with No Subscribers

You can always clear or delete subscriber VLANs manually. However, you can also configure the interface to automatically remove dynamic subscriber VLANs when no client sessions (for example, DHCP or PPPoE) exist on the VLAN.

When configuring automatic removal of dynamic subscriber VLANs, keep the following in mind:

- You can configure automatic VLAN removal only on individual physical interfaces. You cannot configure the feature globally.

- Automatic VLAN removal is not supported for use on Layer 2 Wholesale interfaces. See *Layer 2 and Layer 3 Wholesale Overview*.
- PPPoE subscriber interfaces require the use of a dynamic profiles when configured over dynamic VLANS. However, dynamic profiles are not required for use with DHCP subscriber interfaces that use underlying dynamic VLANs. Because the remove-when-no-subscribers functionality triggers when no dynamic client sessions exist on a dynamic VLAN, automatic removal of underlying dynamic VLANs is not supported when DHCP subscriber interfaces are not created using dynamic profiles.
- The `maintain-subscriber` statement and `remove-when-no-subscribers` statement are mutually exclusive. When the router is configured to maintain subscribers, you cannot also specify that dynamically configured VLAN interfaces are removed when no subscribers exist.
- If PPPoE subscriber session lockout is also configured, the router does not remove the unused subscriber VLAN until the lockout time has expired for each client undergoing lockout on the underlying interface.

To configure automatic removal of subscriber VLANs when no client sessions exist on the VLAN:

1. Access the interface for which you want to enable automatic removal of subscriber VLANs.

```
user@host# edit interfaces ge-1/1/1
```

2. Access the interface automatic configuration hierarchy.

```
[edit interfaces ge-1/1/1]
user@host# edit auto-configure
```

3. Enable subscriber VLAN removal with the `remove-when-no-subscribers` statement.

```
[edit interfaces ge1/1/1 auto-configure]
user@host# set remove-when-no-subscribers
```

RELATED DOCUMENTATION

Dynamic 802.1Q VLAN Overview | 5

Layer 2 and Layer 3 Wholesale Overview

Layer 2 Wholesale Network Topology Overview

PPPoE Subscriber Session Lockout Overview | 240

Verifying and Managing Dynamic VLAN Configuration

IN THIS SECTION

Purpose

View or clear information about dynamic VLANs and stacked VLANs.

Action

- To display subscriber dynamic VLAN information:

```
user@host>show subscribers detail
```

- To display interface-specific output for dynamic VLANs:

```
user@host>show interfaces interface-name
```

- To clear the binding state of dynamic VLAN interfaces:

```
user@host> clear auto-configuration interfaces
```

RELATED DOCUMENTATION

CLI Explorer

CHAPTER 3

Configuring Subscriber Authentication for Dynamic VLANs

IN THIS CHAPTER

Configuring an Authentication Password for VLAN or Stacked VLAN Ranges

You can specify an authentication password for dynamically created VLAN or stacked VLAN interfaces at the `[edit interfaces` *interface-name* `auto-configure vlan-ranges authentication]` or `[edit interfaces` *interface-name* `auto-configure stacked-vlan-ranges authentication]` hierarchy level. This password is sent to the external AAA authentication server for subscriber authentication.

NOTE: You must configure the `username-include (Interfaces)` statement to enable the use of authentication. The `password (Interfaces)` statement is not required and does not cause the interface to use authentication if the `username-include (Interfaces)` statement is not included.

To configure an authentication password:

1. Access the interface over which you want to create dynamic VLANs.

```
user@host# edit interfaces ge-0/0/0
```

2. Edit the VLAN auto-configure stanza.

```
[edit interfaces ge-0/0/0]
user@host# edit auto-configure
```

3. Edit the vlan-ranges or stacked-vlan-ranges stanza.

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit vlan-ranges
```

or

```
[edit interfaces ge-0/0/0 auto-configure]
user@host# edit stacked-vlan-ranges
```

4. Edit the VLAN authentication stanza.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# edit authentication
```

5. Specify a password that is sent to the external AAA authentication server for subscriber authentication.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# set password (Interfaces) $ABC123
```

RELATED DOCUMENTATION

Configuring Dynamic Authentication for VLAN Interfaces | 33

Configuring Dynamic Authentication for VLAN Interfaces

Before you configure dynamic VLAN authentication, configure DHCP Local Server or DHCP Relay over which you want the dynamic VLAN interfaces to function.

For information about DHCP Local Server or DHCP Relay, see:

- *Understanding Differences Between Legacy DHCP and Extended DHCP*
- *Extended DHCP Relay Agent Overview*

> **NOTE**: You can also configure dynamically created VLAN interfaces over PPP or PPPoE interfaces. For information about how to configure PPP or PPPoE, see *Dynamic Profiles for PPP Subscriber Interfaces Overview* or "Subscriber Interfaces and PPPoE Overview" on page 186.

You can use dynamic profiles, in conjunction with RADIUS, to dynamically create logical VLAN interfaces in the default logical system and in a specified routing instance. As DHCP clients in the same VLAN become active, corresponding interfaces are assigned to any specified routing instances. You can also dynamically create an underlying VLAN interface for incoming subscribers, associate interfaces created on this VLAN with the default logical system and a specified routing instance, and define RADIUS authentication values for the dynamically created interfaces.

To configure dynamic authentication for dynamically created VLAN interfaces:

1. Configure an access profile that contains the appropriate accounting order, authentication order, and server access values.

   For information about how to configure an access profile, RADIUS accounting, RADIUS statistics, and how to define RADIUS server access, see:

   - *Configuring Access Profile Options for Interactions with RADIUS Servers*
   - *Specifying the Authentication and Accounting Methods for Subscriber Access*
   - *Configuring Per-Subscriber Session Accounting*
   - *RADIUS Servers and Parameters for Subscriber Access*

2. Configure a dynamic profile that uses the default logical system and creates specific routing instances to contain dynamically created VLAN interfaces.

   See "Configuring a VLAN Dynamic Profile That Associates VLANs with Separate Routing Instances" on page 27.

3. Define the VLAN physical interface for automatic configuration.

   See the following topics:

- Enabling VLAN Tagging
- "Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs" on page 22
- "Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs" on page 18
- "Configuring an Authentication Password for VLAN or Stacked VLAN Ranges" on page 31
- "Configuring VLAN Interface Username Information for AAA Authentication" on page 39

4. Associate an access profile to the VLAN interface.
5. Associate a dynamic profile to the VLAN interface.

RELATED DOCUMENTATION

Dynamic 802.1Q VLAN Overview | 5

Subscriber Packet Type Authentication Triggers for Dynamic VLANs

IN THIS SECTION

By default, VLAN authentication is triggered by any of the packet types specified with the `accept` statement in the dynamic profile that instantiates the VLAN and subscriber interfaces. For certain business cases, you may want a more generic dynamic profile that includes several packet types, but in some situations want the VLAN to be authenticated for only a subset of your customers. You can use the `packet-types` statement to specify the desired subset.

Sample Uses for Packet Type Triggering

The following two use cases describe circumstances when you might want to authenticate a VLAN for only certain subscribers and not others.

- **Conserving resources in a mixed access model**—A mixed access model might employ dynamic VLANs to provide services for PPPoE subscribers, IPoE subscribers, IPv6oE subscribers, or other subscriber types. Typically, the PPPoE subscribers are residential customers, and the IP subscribers are business customers. An understanding of dynamic VLAN authentication and profile instantiation for these subscribers can help you conserve system resources and avoid some impacts to scaling limits.

  By default, authentication is configured for the interface based on the configured VLAN range or stacked VLAN range. Consequently, every dynamic VLAN created in the range must be authenticated, regardless of the packet type that triggers VLAN creation. This works well for the IPoE and IPv6oE subscribers, because dynamic VLAN authentication enables RADIUS-sourced services, such as CoS and filters, to be provisioned. However, the PPPoE subscribers are authenticated by PPP, making the dynamic VLAN authentication unnecessary and a waste of system resources.

  You can avoid this waste by restricting dynamic VLAN authentication to only the VLANs that need it. The `packet-types` statement enables you to specify that only a subset of the packet types accepted on the VLAN interface can trigger authentication. For example, in this heterogeneous access model, the VLAN dynamic profiles accept PPPoE, IPoE, and IPv6oE packets. When you use the `packet-types` statement to specify that only IPoE or IPv6oE packets can initiate VLAN authentication, the PPPoE VLANs are not submitted to RADIUS for authentication.

- **Overriding dynamic profiles in a mixed access model**—Another use for packet-type triggering is to override the configured dynamic profile for certain subscribers. To accomplish this, create one dynamic profile to match the needs of the PPPoE subscribers and create another dynamic profile for the IPoE subscribers. PPPoE subscribers make up the majority of subscribers in this model, so the PPPoE-tuned dynamic profile is applied to the VLAN interface. Include the IP profile in the Juniper Networks Client-Profile-Name VSA [26-174]. Configure the `packet-types` statement to specify that only IP packets trigger VLAN authentication.

  When an IPoE packet is received, RADIUS authenticates the VLAN. RADIUS returns the override profile contained in the Client-Profile-Name VSA and any other session attributes in the Access-Accept message. The VLAN autoconfiguration process overrides the PPPoE profile by instantiating the IP profile for the IPoE subscriber.

Packet Types for VLAN Creation and Authentication

Table 2 on page 36 lists the packet types that you can configure for VLAN authentication depending on the packet types configured for VLAN creation.

Table 2: Relationship Between Packet Types for VLAN Creation and Authentication

Packet Types for VLAN Creation	Packet Types for VLAN Authentication
`any`	Any combination of `any`, `dhcp-v4` or `inet`, `dhcp-v6` or `inet6`, and `pppoe`.
`dhcp-v4`	Either `dhcp-v4` or `inet`.
`dhcp-v6`	Either `dhcp-v6` or `inet6`.
`inet`	Either `dhcp-v4` or `inet`.
`inet6`	Either `dhcp-v6` or `inet6`.
`pppoe`	`pppoe`

NOTE: You cannot simultaneously configure both `dhcp-v4` and `inet` or `dhcp-v6` and `inet6` as packet types for VLAN creation or authentication.

Authentication is performed for all VLANs in either of the following cases:

- You do not specify a packet type to trigger authentication.
- You configure the `any` option for both VLAN creation and authentication.

In general, VLAN authentication is performed when any packet of the type configured to trigger VLAN creation matches one of the packet types configured to trigger VLAN authentication. However, for certain combinations of configured packets, a specific packet is required to trigger authentication. Table 3 on page 36 lists these special cases.

Table 3: Packet Types Required to Trigger Authentication for Special Configuration Combinations

Packet Type for VLAN Creation	Packet Type for VLAN Authentication	Packet Required to Trigger Authentication
`any`	`inet`	any IPv4 packet

Table 3: Packet Types Required to Trigger Authentication for Special Configuration Combinations *(Continued)*

Packet Type for VLAN Creation	Packet Type for VLAN Authentication	Packet Required to Trigger Authentication
`any`	`inet6`	any IPv6 packet
`any`	`dhcp-v4`	DHCP discover
`any`	`dhcp-v6`	DHCPv6 solicit
`dhcp-v4`	`inet`	DHCP discover
`dhcp-v6`	`inet6`	DHCPv6 solicit
`inet`	`dhcp-v4`	DHCP discover
`inet6`	`dhcp-v6`	DHCPv6 solicit

RELATED DOCUMENTATION

Configuring Subscriber Packet Types to Trigger VLAN Authentication | 37

Configuring Subscriber Packet Types to Trigger VLAN Authentication

By default, VLAN authentication is triggered by any of the packet types specified with the `accept` statement in the dynamic profile that instantiates the VLAN and subscriber interfaces. For certain business cases, you may want a more generic dynamic profile that includes several packet types, but in some situations want the VLAN to be authenticated for only a subset of your customers. You can use the `packet-types` statement to specify the desired subset.

To limit triggering of VLAN authentication to a subset of accepted packet types:

- Specify one or more packet types that you want to trigger VLAN authentication.

```
[edit interfaces interface-name auto-configure vlan-ranges authentication]
user@host# set packet-types [packet-type]
```

For example, the following partial configuration shows how to specify that IP, IPv6, and PPPoE packet types trigger the creation of autoconfigured, single-tagged VLANs, but only IP and IPv6 packets trigger authentication:

1. Access the VLAN dynamic profile for which you want to configure VLAN ranges.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges]
user@host# edit dynamic-profile VLAN-PROF-1
```

2. Specify the VLAN ranges for the VLAN dynamic profile.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges dynamic-profile VLAN-PROF-1]
user@host# set ranges any
```

3. Specify the VLAN packet types accepted by the VLAN dynamic profile.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges dynamic-profile VLAN-PROF-1]
user@host# set accept [inet inet6 pppoe]
```

4. Specify the subset of those packet types that you want to trigger VLAN authentication.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication]
user@host# set packet-types [inet inet6]
```

RELATED DOCUMENTATION

Configuring a Dynamic Profile Used to Create Single-Tag VLANs | 15

Configuring a Dynamic Profile Used to Create Stacked VLANs | 19

Configuring VLAN Interface Username Information for AAA Authentication

You can define interface information that is included in the username that is subsequently passed to the external AAA authentication service (for example, RADIUS) when creating dynamic VLANs or stacked VLANs. The AAA authentication service uses this information to authenticate the VLAN or stacked VLAN physical interface. After the interface is authenticated, the AAA service can send the required routing instance values to the system for use in dynamically creating VLAN or stacked VLAN interfaces.

NOTE: The following example configures username information on VLANs. However, you can also configure dynamic authentication on stacked VLANs by configuring the same statements at the `[edit interfaces` *`interface-name`* `auto-configure stacked-vlan-ranges authentication]` hierarchy level.

To configure VLAN interface username information:

1. Access the `authentication` stanza for the interface over which you want to configure username information.

   ```
   user@host# edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication
   ```

2. Specify the username components that you want the AAA authentication service to use to authenticate the username.

   - Include the agent circuit identifier (ACI). The ACI is conveyed by the Access-Loop-Circuit-ID TLV in an out-of-band ANCP Port Up message.

     ```
     [edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
     user@host# set username-include circuit-id
     ```

   - Include the circuit type.

     ```
     [edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
     user@host# set username-include circuit-type
     ```

- Specify the character used as the delimiter between the concatenated components of the username.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include delimiter delimiter-character
```

- Specify the domain name that is concatenated with the username.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include domain-name domain-name-string
```

- Include the interface name and VLAN tags.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include interface-name
```

- Include the client hardware address (chaddr) from the incoming DHCP discover packet.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include mac-address
```

- Include the option 18 (Interface-ID) information that was received in the innermost DHCPv6 Relay-Forward message header.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include option-18
```

- Include the option 37 (DHCPv6 Relay Agent Remote-ID) information that was received in the innermost DHCPv6 Relay-Forward message header.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include option-37
```

- Include the option 82 information from the client PDU. For DHCPv4, optionally include suboption 1 (Agent Circuit ID) or suboption 2 (Agent Remote ID).

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include option-82 <circuit-id> <remote-id>
```

- Include the user-defined RADIUS realm string to direct the authentication request to a profile that does not allocates addresses.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include radius-realm radius-realm-string
```

- Include the agent remote identifier (ARI). The ARI is conveyed by the Access-Loop-Remote-ID TLV in an out-of-band ANCP Port Up message

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include remote-id
```

- Specify a user prefix.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include user-prefix user-prefix-string
```

- Include the subscriber VLAN tags. You can use this option instead of the `interface-name` option when the outer VLAN tag is unique across the system and you do not need the underlying physical interface name to be part of the format.

```
[edit interfaces ge-0/0/0 auto-configure vlan-ranges authentication username-include]
user@host# set username-include vlan-tags
```

RELATED DOCUMENTATION

Configuring Dynamic Authentication for VLAN Interfaces | 33

Using DHCP Option 82 Suboptions in Authentication Usernames for Autosense VLANs | 42

Using DHCP Option 82 Suboptions in Authentication Usernames for Autosense VLANs

You can specify the Option 82 suboptions that are concatenated with the username during the authentication process for autosense VLANs. The option 82 value used in creating the username is based on the option 82 value that is encoded in the incoming DHCP discover packet.

You can specify either, both, or neither of the Agent Circuit ID (suboption 1) and the Agent Remote ID (suboption 2). If you specify both, the Agent Circuit ID is supplied first, followed by a delimiter, and then the Agent Remote ID. If you specify that neither suboption is supplied, the raw payload of Option 82 from the PDU is concatenated to the username. The use of Option 82 suboptions is supported for DHCPv4 discover packets only.

RELATED DOCUMENTATION

Configuring VLAN Interface Username Information for AAA Authentication | 39

Using DHCP Option 18 and Option 37 in Authentication Usernames for DHCPv6 Autosense VLANs

For DHCPv4, Option 82 has suboptions containing the ACI and ARI that are concatenated with the username during the authentication process for autosense (dynamic) VLANs. For DHCPv6, the relay agent uses Options 18 and Option 37 to convey the ACI and ARI, respectively. You can include these options in the username to generate unique usernames that identify subscribers for authentication in DHCPv6 dynamic VLANs.

A DHCPv6 Solicit message encapsulated with a Relay-Forward message header and one without the Relay-Forward message header are eligible for dynamic VLAN creation when you configure the DHCPv6 packet type for autosensing. Options 18 and Option 37 are provided in the Relay-Forward message header and are extracted only from this header and not from the options within the DHCPv6 Solicit message. In addition, if the DHCPv6 Solicit message is encapsulated in multiple Relay-Forward message headers, only the option values from the innermost Relay-Forward message header are used for username authentication. If these options are sent by the client or DHCPv6 relay agent, and if dynamic VLAN authentication is configured to use these options in the username, then the options are included in the username for authentication. If either of these options is not sent by the client or DHCPv6 relay agent, or if the dynamic VLAN authentication is not configured to use the option in the username, the username is constructed without the option.

To include Option 18 or Option 37 in the username for DHCPv6 dynamic VLANs, include the `option-37` and `option-18` statements at the `[edit interfaces interface-name auto-configure vlan-ranges authentication username-include]` hierarchy level. To include Options 18 or Option 37 in the username for stacked VLANs, include `option-18` and `option-37` statements at the `[edit interfaces interface-name auto-configure stacked-vlan-ranges authentication username-include]` hierarchy level.

RELATED DOCUMENTATION

Configuring VLAN Interface Username Information for AAA Authentication | 39

CHAPTER 4

Configuring VLANs for Households or Individual Subscribers Using ACI-Based Dynamic VLANs

IN THIS CHAPTER

Agent Circuit Identifier-Based Dynamic VLANs Overview

IN THIS SECTION

Dynamic VLAN subscriber interfaces that are created based on the agent circuit identifier (ACI) value are useful in configurations with a mix of DHCP and PPPoE subscriber sessions at the same household.

When you use service VLANs (S-VLANs) to carry one service to many subscribers (1:N), each subscriber or household can have different types of traffic on multiple VLANs. To identify all subscriber sessions for an individual subscriber or a household, you can use the value of the ACI string. The ability to uniquely identify subscribers simplifies the application of services, such as CoS and filters, to individual subscribers or households.

Because an S-VLAN corresponds to a service rather than an individual subscriber, the router uses ACI information in DHCP and PPPoE control packets instead of VLAN encapsulation to uniquely identify subscribers and facilitate application of subscriber-based services.

ACI VLANs and ALI VLANs

The legacy ACI method for configuring the creation of dynamic VLANs is based on the receipt of only the ACI. When the ACI is not received, no VLAN is created. An alternative method provides greater flexibility than the legacy method. The access-line-identifier (ALI) method enables dynamic VLANs to be created based on receipt of the ACI, the agent remote identifier (ARI), both the ACI and the ARI, or the absence of both of ACI and ARI.

Although the agent circuit identifier is also an access-line identifier, we use specific terminology to distinguish between the two configuration methods:

- The documentation continues to use the terms *agent circuit identifier*, *ACI*, and *ACI-based* to refer only to VLANs and interface sets configured with the legacy method, using the `agent-circuit-identifier` stanza for autoconfiguration.
- The documentation uses the terms *access-line identifier*, *ALI*, and *ALI-based* to refer to VLANs and interface sets configured with the access-line-identifier method, using the `line-identity` stanza for autoconfiguration.

You must configure only one of these methods. A CLI check prevents you from configuring both of these methods. You can use the ALI method to achieve the same results as the legacy ACI method. Apart from the fact that the ALI method uses the `line-identity` stanza instead of the `agent-circuit-identifier` stanza for autoconfiguration, the configuration is the same for both methods. The legacy ACI method might be deprecated in the future in favor of the more generic ALI method. For information about ALI VLANs, see "Access-Line-Identifier-Based Dynamic VLANs Overview" on page 61.

How ACI-Based Dynamic VLANs Work

The process for creating an ACI-based dynamic VLAN is as follows:

1. The residential gateway at a household sends a connection request to the access node.

2. The access node identifies the household and inserts an ACI value into the header of a DHCP or PPPoE control packet. The access node can insert the ACI value into one of the following DHCP options or PPPoE control packets:

   - Option 82 of DHCP packets
   - Option 18 of DHCPv6 packets
   - The DSL Forum Agent-Circuit-ID VSA [26-1] (option 0x105) of PPPoE Active Discovery Initiation (PADI) and PPPoE Active Discovery Request (PADR) control packets

   The access node inserts the same ACI value to all subsequent sessions that originate from the same household.

3. The access node forwards the control packets to the BNG.

4. When the BNG receives the control packets, it extracts the ACI value in the header and uses it to build a unique dynamic VLAN subscriber interface.

   Subsequent control traffic sent from the same household will contain the same ACI value. The BNG groups subscriber interfaces that have the same ACI value into an ACI interface set, also called an ACI set.

The BNG can then apply CoS and policies to the ACI set to dynamically provision traffic for a household.

Interface Hierarchy When ACI Interface Sets Are Used

The following describes the components of an ACI-based dynamic VLAN configuration, from bottom to top of the interface stack:

Static Physical Interface

ACI-based dynamic VLAN configurations support the following physical interface types:

- Gigabit Ethernet
- Aggregated Ethernet

You can configure ACI-based dynamic VLAN subscriber interfaces on Modular Port Concentrators/Modular Interface Cards (MPCs/MICs) that face the access side of the network in an MX Series router.

Underlying VLAN Interface

After you define the ACI interface set, you must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on ACI information. You can configure the underlying VLAN interface either dynamically (with a dynamic profile) or statically.

ACI-based dynamic VLAN configurations support the following underlying VLAN interface types:

- Gigabit Ethernet
- VLAN demux (demux0)

> **NOTE**: When you configure an underlying VLAN interface to support creation of ACI-based dynamic VLANs, we recommend that you use this underlying interface only for subscriber interfaces that contain agent-circuit-identifier information in their DHCP or PPPoE control packets. If the router receives DHCP or PPPoE control packets without agent-circuit-identifier information on an underlying VLAN interface configured for ACI-based dynamic VLANs, the associated subscriber interfaces might not instantiate successfully.

Dynamic ACI Interface Set

The dynamic ACI interface set groups the DHCP and PPPoE subscriber sessions that belong to a particular household and share a common unique ACI value. The router creates one ACI interface set per household.

You must create a dynamic profile to define the ACI interface set, which is represented in the profile by the Junos OS predefined dynamic variable `$junos-interface-set-name`. When a DHCP or PPPoE subscriber accesses the router on a particular interface, the router obtains the agent-circuit-identifier information from the DHCP or PPPoE control packets transmitted on that interface and dynamically creates the ACI interface set when the first subscriber from that household logs in.

ACI-Based Dynamic Subscriber Interface

You must create a dynamic profile to define either a dynamic PPPoE subscriber interface for PPPoE subscriber sessions, or a dynamic IP demultiplexer (IP demux) subscriber interface for DHCP subscriber sessions. The router creates the subscriber interface when a subscriber logs in on the associated underlying VLAN interface associated with the dynamic profile that defines the ACI interface set.

RELATED DOCUMENTATION

Subscriber Management VLAN Architecture Overview | 2
Configuring Dynamic VLANs Based on Agent Circuit Identifier Information | 48
Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57
Clearing Agent Circuit Identifier Interface Sets | 59

Configuring Dynamic VLANs Based on Agent Circuit Identifier Information

You can configure dynamic VLAN subscriber interfaces based on agent circuit identifier (ACI) information, also known as *ACI-based dynamic VLANs*, for DHCP and PPPoE subscribers. To do so, you create an *ACI interface set*, which is a logical collection of subscriber interfaces that originate at the same household or on the same access-loop port, and then reference the ACI interface set in the dynamic profile for a PPPoE or IP demultiplexing (IP demux) logical subscriber interface.

Before you begin:

1. Configure the underlying physical interface for single-tag VLANs or stacked (dual-tag) VLANs.

   See the following topics:

   - "Configuring a Dynamic Profile Used to Create Stacked VLANs" on page 19
   - "Configuring a Dynamic Profile Used to Create Single-Tag VLANs" on page 15
   - "Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs" on page 18
   - "Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs" on page 22

2. Create a dynamic profile that defines the logical subscriber interface.

   See the following topics:

   - *Configuring a Basic Dynamic Profile*
   - "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194

To configure a dynamic VLAN subscriber interface based on ACI information:

1. Configure a dynamic profile that defines the dynamic ACI interface set.

   See "Defining ACI Interface Sets" on page 49.

2. (Optional) In the dynamic profile for the ACI interface set, configure the router to use the Actual-Data-Rate-Downstream VSA [26-130] or Access-Loop-Encapsulation VSA [26-144] value in PPPoE control packets to adjust CoS shaping-rate and overhead-accounting attributes at a per-household level.

   See *Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs*.

3. Dynamically or statically configure the underlying VLAN logical interface to enable dynamic subscriber interface creation based on ACI information.

- For dynamic underlying VLAN interfaces, see "Configuring Dynamic Underlying VLAN Interfaces to Use Agent Circuit Identifier Information" on page 51.

- For static underlying VLAN interfaces, see "Configuring Static Underlying VLAN Interfaces to Use Agent Circuit Identifier Information" on page 53.

4. Associate the dynamic ACI interface set with the dynamic PPPoE or dynamic IP demux logical subscriber interface.

   See "Configuring Dynamic VLAN Subscriber Interfaces Based on Agent Circuit Identifier Information" on page 54.

5. (Optional) In the dynamic profile for the PPPoE (`pp0`) subscriber interface, configure the router to use the Actual-Data-Rate-Downstream VSA [26-130] or Access-Loop-Encapsulation VSA [26-144] value in PPPoE control packets to adjust CoS shaping-rate and overhead-accounting attributes at a per-subscriber level.

   See *Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs*.

RELATED DOCUMENTATION

Defining ACI Interface Sets

To configure the router to create dynamic VLAN subscriber interfaces for DHCP and PPPoE subscribers based on ACI information, you must create a dynamic ACI interface set.

To configure an ACI interface set in a dynamic profile:

1. Access the dynamic profile that defines the ACI interface set.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Configure the dynamic ACI interface set.

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

   Use the `$junos-interface-set-name` predefined variable to represent the name of the ACI interface set. It is replaced with the actual ACI interface set name generated by the router when the first subscriber from that household logs in.

3. Include the underlying interfaces for the dynamic ACI interface set.

```
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”]
user@host# set interface $junos-interface-ifd-name
```

   Use the `$junos-interface-ifd-name` predefined variable to represent the name of the interface. The variable is replaced with the name of the interface on which the subscriber accesses the BNG.

   The `unit` statement is not required in the dynamic profile when you configure an ACI interface set.

4. (Optional) For dynamic PPPoE subscriber interfaces, configure the maximum number of dynamic PPPoE sessions that the router can activate for the ACI interface set; that is, for the same household.

```
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”]
user@host# edit pppoe-underlying-options
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”
pppoe-underlying-options]
user@host# set max-sessions number
```

5. (Optional) Apply attributes for CoS and interface filters to all subscriber interfaces belonging to the ACI interface set.

The following example shows the minimum dynamic profile required to define an ACI interface set named `aci-vlan-set-profile`. It uses predefined variables to represent the interface set and the underlying physical interface.

```
[edit dynamic-profiles aci-vlan-set-profile]
interfaces {
    interface-set "$junos-interface-set-name" {
        interface "$junos-interface-ifd-name";
    }
}
```

RELATED DOCUMENTATION

Agent Circuit Identifier-Based Dynamic VLANs Overview | 44

Configuring Dynamic VLANs Based on Agent Circuit Identifier Information | 48

Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57

Clearing Agent Circuit Identifier Interface Sets | 59

Applying CoS Attributes to VLANs Using Agent-Circuit-Identifiers

Example: Implementing a Filter for Households That Use ACI-Based VLANs

Configuring Dynamic Underlying VLAN Interfaces to Use Agent Circuit Identifier Information

After you define the agent circuit identifier (ACI) interface set, you must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on ACI information. You can configure the underlying VLAN interface statically or dynamically.

This topic describes how to configure the underlying VLAN interface *dynamically*.

Before you begin:

- Create a dynamic profile that defines the underlying VLAN interface.

  See the following topics:

  - *Configuring a Basic Dynamic Profile*
  - "Configuring a Dynamic Profile Used to Create Single-Tag VLANs" on page 15
  - "Configuring a Dynamic Profile Used to Create Stacked VLANs" on page 19

To configure a dynamic underlying VLAN interface to use ACI information:

- In the dynamic profile for the underlying VLAN interface, associate the dynamic profile that defines the ACI interface set with the underlying VLAN interface.

  ```
  [edit dynamic-profiles profile-name]
  user@host# set interfaces interface-name unit logical-unit-number auto-configure agent-
  circuit-identifier dynamic-profile aci-interface-set-profile-name
  ```

  For example, the following statement in a dynamic profile named aci-vlan-underlying-profile-demux associates the dynamic underlying VLAN interface with dynamic profile aci-vlan-set-profile2 that defines the ACI interface set. You must use the predefined dynamic variable `$junos-interface-ifd-name` to represent the interface name, and `$junos-interface-unit` to represent the logical unit number.

```
[edit dynamic-profiles aci-vlan-underlying-profile-demux]
user@host# set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit” auto-
configure agent-circuit-identifier dynamic-profile aci-vlan-set-profile2
```

The following example shows the dynamic configuration that uses this statement. This configuration enables the underlying dynamic IP demultiplexing (IP demux) VLAN interface to create dynamic subscriber interfaces based on ACI information by applying a single default ACI interface set dynamic profile (aci-vlan-set-profile2) to all households on the VLAN interface.

```
[edit dynamic-profiles aci-vlan-underlying-profile-demux]
interfaces {
    “$junos-interface-ifd-name” {
        unit "$junos-interface-unit" {
            auto-configure {
                agent-circuit-identifier {
                    dynamic-profile aci-vlan-set-profile2;
                }
            }
            vlan-id "$junos-vlan-id";
            demux-options {
                underlying-interface “$junos-interface-ifd-name”;
            }
            family inet {
                unnumbered-address lo0.0 preferred-source-address 198.51.100.20;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Agent Circuit Identifier-Based Dynamic VLANs Overview | 44
Configuring Dynamic VLANs Based on Agent Circuit Identifier Information | 48
Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57

Configuring Static Underlying VLAN Interfaces to Use Agent Circuit Identifier Information

After you define the agent circuit identifier (ACI) interface set, you must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on ACI information. You can configure the underlying VLAN interface statically or dynamically.

This topic describes how to configure the underlying VLAN interface statically.

To configure a static underlying VLAN interface to use ACI information:

- Associate the dynamic profile that defines the ACI interface set with the static underlying VLAN interface.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number auto-configure agent-
circuit-identifier dynamic-profile aci-interface-set-profile-name
```

  For example, the following statement associates static Gigabit Ethernet VLAN interface ge-1/0/0.0 with the dynamic profile aci-vlan-set-profile that defines the ACI interface set.

```
[edit]
user@host# set interfaces ge-1/0/0 unit 0 auto-configure agent-circuit-identifier dynamic-
profile aci-vlan-set-profile
```

The following example shows the static configuration that uses this statement. This configuration enables the underlying VLAN interface ge-1/0/0.0 to create dynamic subscriber interfaces based on ACI information by applying a single default ACI interface set dynamic profile (aci-vlan-set-profile) to all households on the VLAN interface.

```
[edit]
interfaces {
    ge-1/0/0 {
        flexible-vlan-tagging;
        unit 0 {
            vlan-id 100;
            auto-configure {
                agent-circuit-identifier {
                    dynamic-profile aci-vlan-set-profile;
                }
            }
```

```
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic VLAN Subscriber Interfaces Based on Agent Circuit Identifier Information

After you define the dynamic agent circuit identifier (ACI) interface set and enable creation of ACI-based dynamic VLAN subscriber interfaces on the underlying VLAN interface, you must complete the configuration by associating the ACI interface set with the PPPoE or IP demultiplexing (IP demux) subscriber interface in the dynamic profile for the subscriber interface.

Before you begin:

- Create a dynamic profile that defines the logical subscriber interface.

  See the following topics:

  - *Configuring a Basic Dynamic Profile*
  - "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194

To configure a dynamic VLAN subscriber interface based on ACI information:

- In the dynamic profile for the PPPoE or IP demux subscriber interface, associate the dynamic ACI interface set with the dynamic VLAN subscriber interface name (`pp0` or `demux0`) and logical unit number.

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

  For example, the following statement in a dynamic profile named aci-vlan-pppoe-profile associates the dynamic ACI interface set with the dynamic `pp0` (PPPoE) logical subscriber interface. You must use the predefined dynamic variable `$junos-interface-set-name` to represent the name of the dynamic ACI interface set, and `$junos-interface-unit` to represent the logical unit number of the subscriber interface.

```
[edit dynamic-profiles aci-vlan-pppoe-profile]
user@host# set interfaces interface-set $junos-interface-set-name interface pp0 unit $junos-
interface-unit
```

Similarly, the following statement in a dynamic profile named aci-vlan-demux-profile associates the dynamic ACI interface set (represented by `$junos-interface-set-name`) with the `demux0` (IP demux) logical subscriber interface.

```
[edit dynamic-profiles aci-vlan-demux-profile]
user@host# set interfaces interface-set $junos-interface-set-name interface demux0 unit
$junos-interface-unit
```

The following examples show the dynamic configurations that use each of these statements. The following sample configuration shows a dynamic profile named aci-vlan-pppoe-profile for an ACI-based dynamic PPPoE (`pp0`) subscriber interface for use by PPPoE subscribers.

```
[edit dynamic-profiles aci-vlan-pppoe-profile]
interfaces {
    interface-set "$junos-interface-set-name" {
        interface pp0 {
            unit "$junos-interface-unit";
        }
    }
    pp0 {
        unit "$junos-interface-unit" {
            ppp-options {
                chap;
                pap;
            }
            pppoe-options {
                underlying-interface "$junos-underlying-interface";
                server;
            }
            no-keepalives;
            family inet {
                unnumbered-address lo0.0;
            }
```

```
        }
    }
}
```

The following sample configuration shows a dynamic profile named aci-vlan-demux-profile for an ACI-based dynamic IP demux(`demux0`) subscriber interface for use by DHCP subscribers.

```
[edit dynamic-profiles aci-vlan-demux-profile]
interfaces {
    interface-set "$junos-interface-set-name" {
        interface demux0 {
            unit "$junos-interface-unit";
        }
    }
    demux0 {
        unit "$junos-interface-unit" {
            demux-options {
                underlying-interface "$junos-underlying-interface";
            }
            family inet {
                demux-source {
                    $junos-subscriber-ip-address;
                }
                unnumbered-address lo0.0 preferred-source-address  198.51.100.202;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Agent Circuit Identifier-Based Dynamic VLANs Overview | 44

Configuring Dynamic VLANs Based on Agent Circuit Identifier Information | 48

Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57

Clearing Agent Circuit Identifier Interface Sets | 59

Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration

IN THIS SECTION

Purpose

View information about dynamic agent circuit identifier (ACI) interface sets and ACI-based dynamic VLAN subscriber interfaces configured on the router.

Action

- To display the logical and physical interface associations for the classifier, rewrite rules, scheduler map objects, and CoS adjustment settings:

```
user@host> show class-of-service interface interface-name
```

- To display the CoS associations for the specified dynamic ACI interface set:

```
user@host> show class-of-service interface-set aci-interface-set-name
```

- To display information about the specified CoS traffic shaping and scheduling profile:

```
user@host> show class-of-service traffic-control-profile profile-name
```

- To display address bindings and ACI interface set information in the client table on the extended DHCP local server:

```
user@host> show dhcp server binding detail
```

- To display status information about a specified Gigabit Ethernet interface:

```
user@host> show interfaces ge-fpc/pic/port.logical-unit-number
```

- To display status information about a specified IP demultiplexing (IP demux) interface:

```
user@host> show interfaces demux0.logical-interface-number
```

- To display information about all dynamic ACI interface sets configured on the router:

```
user@host> show interfaces interface-set
```

- To display session-specific information about ACI-based dynamic PPPoE subscriber interfaces:

```
user@host> show pppoe interfaces pp0.logical-unit-number
```

- To display information about PPPoE underlying interfaces, including whether creation of ACI-based dynamic VLAN subscriber interfaces is enabled on the underlying interface:

```
user@host> show pppoe underlying-interfaces logical-interface-name detail
```

- To display information about active subscriber sessions associated with ACI interface sets:

```
user@host> show subscribers detail
```

- To display information about active subscriber sessions associated with a specified ACI interface set:

```
user@host> show subscribers aci-interface-set-name aci-interface-set-name detail
```

- To display information about active subscriber sessions that have an agent circuit identifier value containing a matching substring:

```
user@host> show subscribers agent-circuit-identifier agent-circuit-identifier-substring detail
```

RELATED DOCUMENTATION

Clearing Agent Circuit Identifier Interface Sets

IN THIS SECTION

Purpose

Clear a specified dynamic agent circuit identifier (ACI) interface set configured on the router.

Action

- To clear a specified ACI interface set that has no active members:

```
user@host> clear auto-configuration interfaces interface-set interface-set-name
```

  For example, the following command clears the ACI interface set named aci-1003-ge-1/0/0.4001:

```
user@host> clear auto-configuration interfaces interface-set aci-1003-ge-1/0/0.4001
Interface-set aci-1003-ge-1/0/0.4001 deleted
```

Meaning

The router dynamically creates an ACI interface set, if configured, when the first DHCP or PPPoE subscriber from a particular household logs in. However, the router does not automatically delete the

ACI interface set when the last subscriber from that household logs out. As a result, you must use the `clear auto-configuration interfaces interface-set` command to explicitly clear the ACI interface set when it no longer has any active subscriber interface members. If you attempt to clear an ACI interface that still has active member interfaces, the router displays an error message and rejects the command.

When you specify the name of the ACI interface set to be cleared, you must use the ACI interface set name internally generated by the router, and not the actual ACI string carried in DHCP and PPPoE control packets. The router uses the following format to name ACI interface sets, as shown in the ACI interface set named aci-1003-ge-1/0/0.4001:

`aci-`*`nnnn-interface-name.logical-unit-number`*

where:

- *nnnn* is a randomly generated 4-digit identifier (1003 in the example)
- *interface-name* is the name of the dynamic subscriber interface (ge-1/0/0 in the example)
- *logical-unit-number* is the logical unit number of the dynamic subscriber interface (4001 in the example)

To view the names of the ACI interface sets configured on the router, use the `show subscribers` command.

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Agent Circuit Identifier Information | 48
Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration | 57
CLI Explorer

CHAPTER 5

Configuring VLANs for Households or Individual Subscribers Using Access-Line-Identifier Dynamic VLANs

IN THIS CHAPTER

Access-Line-Identifier-Based Dynamic VLANs Overview

IN THIS SECTION

Dynamic VLAN subscriber interfaces that are created based on the access-line identifier (ALI) are useful in configurations with a mix of DHCP and PPPoE subscriber sessions at the same household. Each household is assigned a unique ALI.

When you use service VLANs (S-VLANs) or N:1 VLANs to carry many subscribers (1:N), each subscriber or household can have different types of traffic on that VLAN. When a Layer 2 Residential Gateway is used, a household may have multiple sessions, any of which can be DHCP or PPPoE. The access node embeds the ALI in DHCP and PPPoE control packets. To identify all subscriber sessions for a household, you can use the ALI. The ability to uniquely identify subscribers simplifies the application of services, such as CoS and filters, to individual subscribers or households.

Because an S-VLAN corresponds to a common broadcast domain rather than an individual household, the router uses the ALI in DHCP and PPPoE control packets instead of VLAN encapsulation to uniquely identify sessions for the same household and facilitate application of subscriber-based services. ALIs include the agent circuit identifier (ACI) and the agent remote identifier (ARI).

ALI VLANs and ACI VLANs

The ALI method for configuring the creation of dynamic VLANs is based on the receipt of a configured trusted option, which can be the ACI, the ARI, both the ACI and the ARI, or the absence of both of ACI and ARI. In case both ACI and ARI are absent, you can configure the dynamic VLANs to be created based on a single network-wide unique MAC address assigned to each household or access line. Another method, called the legacy ACI method, enables dynamic VLANs to be created based only on the ACI. When the legacy method is used and the ACI is not received, no VLAN is created. The ALI method provides greater flexibility than the legacy method; for example, it can be used when the access node embeds only the ARI instead of the ACI.

Although the agent circuit identifier is also an access-line identifier, we use specific terminology to distinguish between the two configuration methods:

- The documentation continues to use the terms *agent circuit identifier*, *ACI*, and *ACI-based* to refer only to VLANs and interface sets configured with the legacy method, using the `agent-circuit-identifier` stanza for autoconfiguration.

- The documentation uses the terms *access-line identifier*, *ALI*, and *ALI-based* to refer to VLANs and interface sets configured with the access-line-identifier method, using the `line-identity` stanza for autoconfiguration.

You must configure only one of these methods. A CLI check prevents you from configuring both of these methods. You can use the ALI method to achieve the same results as the legacy ACI method. Apart from the fact that the ALI method uses the `line-identity` stanza instead of the `agent-circuit-identifier` stanza for autoconfiguration, the configuration is the same for both methods. The ALI method is a more generalized and flexible approach to the legacy ACI method and is therefore recommended. For information about ACI VLANs, see "Agent Circuit Identifier-Based Dynamic VLANs Overview" on page 44.

How ALI-Based Dynamic VLANs Work

The process for creating an ALI-based dynamic VLAN is as follows:

1. The residential gateway at a household sends a connection request to the access node.

2. The access node identifies the household and inserts an access-line-identifier value into the header of a DHCP or PPPoE control packet. The access-line identifier can be the ACI value, the ARI value, or both.Table 4 on page 63 lists where the access node can insert the ALI value for DHCP, DHCPv6, and PPPoE control packets.

   Table 4: Location of the Access-Line Identifier in DHCP, DHCPv6, and PPPoE Control Packets

	DHCP Discover Packets	DHCPv6 Solicit Packets	PPPoE Active Discovery Initiation (PADI) and PPPoE Active Discovery Request (PADR) Control Packets
ACI	Option 82, suboption 1	Option 18	Vendor-Specific tag for DSL-Forum vendor-id 3561, sub-tag 1
ARI	Option 82, suboption 2	Option 37	Vendor-Specific tag for DSL-Forum vendor-id 3561, sub-tag 2

   The access node inserts the same ALI value into the control packets for all subsequent sessions that originate from the same household.

   - When neither the ACI nor the ARI is received and `accept-no-ids` is configured as the line identity trusted option, then the router creates the interface set using an internally generated default string as the identifier value. It creates one such interface set for each underlying logical interface.

   - For PPPoE subscribers you can configure the `accept-no-ids mac-address` to use a single network-wide unique MAC address assigned to each household or access line to create the ALI based dynamic VLAN when neither ACI nor the ARI is received in the PPPoE control packet. When a household has multiple PPPoE sessions, the AN performs MAC address translation such that all PPPoE sessions from a household have a unique source MAC address. To allow multiple PPPoE sessions from the same source MAC address, both PPPoE duplicate-protection and short-cycle-protection support is extended to include `relay-session-id` tag along with source MAC to uniquely identify each PPPoE session.

3. The access node forwards the control packets to the broadband network gateway (BNG).

4. When the BNG receives the control packets, it extracts the ALI value in the header and uses this value to build a unique dynamic VLAN subscriber interface.

Subsequent control traffic sent from the same household contains the same ALI value. The BNG groups subscriber interfaces that have the same ALI value into an ALI interface set, also called an ALI set.

The BNG can then apply CoS and policies to the ALI set to dynamically provision traffic for a household.

Interface Hierarchy When ALI Interface Sets Are Used

The following sections describe the components of an ALI-based dynamic VLAN configuration, from bottom to top of the interface stack.

Static Physical Interface

ALI-based dynamic VLAN configurations support the following physical interface types:

- Gigabit Ethernet
- Aggregated Ethernet
- Pseudowire Subscriber (PS) interfaces used for Pseudowire Headend Termination (PWHT)

You can configure ALI-based dynamic VLAN subscriber interfaces on Modular Port Concentrators/ Modular Interface Cards (MPCs/MICs) that face the access side of the network in an MX Series router.

Underlying VLAN Interface

You must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on the ALI. You can configure the underlying VLAN interface either dynamically (with a dynamic profile) or statically.

ALI-based dynamic VLAN configurations support the following underlying VLAN interface types:

- Gigabit Ethernet
- Pseudowire Subscriber (PS) interfaces used for Pseudowire Headend Termination (PWHT)
- VLAN demux (demux0)

> **NOTE**: If you configure an underlying VLAN interface to support creation of ALI-based dynamic VLANs, we recommend that you use this underlying interface only for subscriber interfaces that contain ALI information in their DHCP or PPPoE control packets. If the router receives DHCP or PPPoE control packets without this information on an underlying VLAN interface configured for ALI-based dynamic VLANs, the

associated subscriber interfaces might not instantiate successfully. The exception to this behavior is when you have configured `accept-no-ids` as the trusted option.

Dynamic ALI Interface Set

The dynamic ALI interface set groups the DHCP and PPPoE subscriber sessions that belong to a particular household and share the same unique ALI value. The router creates one ALI interface set for each household.

You must create a dynamic profile that defines the ALI interface set. The interface set is represented in the profile by the predefined dynamic variable `$junos-interface-set-name`. When a DHCP or PPPoE subscriber accesses the router on a particular interface, the router obtains the ALI from the DHCP or PPPoE control packets transmitted on that interface. If the ALI matches the configured trusted option, the router dynamically creates the ALI interface set when the first subscriber from that household logs in.

ALI-Based Dynamic Subscriber Interface

You must create a dynamic profile to define either a dynamic PPPoE subscriber interface for PPPoE subscriber sessions, or a dynamic IP demultiplexed (IP demux) subscriber interface for DHCP subscriber sessions. The router creates the subscriber interface when a subscriber logs in on the associated underlying VLAN interface associated with the dynamic profile that defines the ALI interface set.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
25.2R1	For PPPoE subscribers you can configure the `line-identity includes accept-no-ids mac-address` to use a single network-wide unique MAC address assigned to each household or access line to create the ALI based dynamic VLAN when neither ACI nor the ARI is received in the PPPoE control packet.

RELATED DOCUMENTATION

Subscriber Management VLAN Architecture Overview | 2
Configuring Dynamic VLANs Based on Access-Line Identifiers | 66

Configuring Dynamic VLANs Based on Access-Line Identifiers

You can configure dynamic VLAN subscriber interfaces for DHCP and PPPoE subscribers based on the access-line identifier (ALI). These subscriber interfaces are also known as *access-line identifier VLANs*, *ALI-based dynamic VLANs*, or *ALI dynamic VLANs*. To configure these VLANs, you create an *ALI interface set*, which is a logical collection of subscriber interfaces that originate at the same household or on the same access-loop port, and then you reference the ALI interface set in the dynamic profile for a PPPoE or IP demultiplexing (IP demux) logical subscriber interface.

Before you begin:

1. Configure the underlying physical interface for single-tag VLANs or stacked (dual-tag) VLANs.

   See the following topics:

   - "Configuring a Dynamic Profile Used to Create Stacked VLANs" on page 19
   - "Configuring a Dynamic Profile Used to Create Single-Tag VLANs" on page 15
   - "Configuring an Interface to Use the Dynamic Profile Configured to Create Single-Tag VLANs" on page 18
   - "Configuring an Interface to Use the Dynamic Profile Configured to Create Stacked VLANs" on page 22

2. Create a dynamic profile that defines the logical subscriber interface.

   See the following topics:

   - *Configuring a Basic Dynamic Profile*
   - "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194

To configure a dynamic VLAN subscriber interface based on the ALI:

1. Configure a dynamic profile that defines the dynamic ALI interface set.

   See "Defining Access-Line-Identifier Interface Sets" on page 67.
2. (Optional) In the dynamic profile for the ALI interface set, configure the router to use the Actual-Data-Rate-Downstream VSA [26-130] or Access-Loop-Encapsulation VSA [26-144] value in PPPoE control packets to adjust CoS shaping-rate and overhead-accounting attributes at a per-household level.

See *Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers*.

3. Dynamically or statically configure the underlying VLAN logical interface to enable dynamic subscriber interface creation based on the ALI.

    - For dynamic underlying VLAN interfaces, see "Configuring Dynamic Underlying VLAN Interfaces to Use Access-Line Identifiers" on page 69.

    - For static underlying VLAN interfaces, see "Configuring Static Underlying VLAN Interfaces to Use Access-Line Identifiers" on page 71.

4. Associate the dynamic ALI interface set with the dynamic PPPoE or dynamic IP demux logical subscriber interface.

    See "Configuring Dynamic VLAN Subscriber Interfaces Based on Access-Line Identifiers" on page 73.

5. (Optional) In the dynamic profile for the PPPoE (`pp0`) subscriber interface, configure the router to use the Actual-Data-Rate-Downstream VSA [26-130] or Access-Loop-Encapsulation VSA [26-144] value in PPPoE control packets to adjust CoS shaping-rate and overhead-accounting attributes at a per-subscriber level.

    See *Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers*.

RELATED DOCUMENTATION

Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers | 76

Clearing Access-Line-Identifier Interface Sets | 78

Access-Line-Identifier-Based Dynamic VLANs Overview | 61

Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers

Defining Access-Line-Identifier Interface Sets

To configure the router to create dynamic VLAN subscriber interfaces for DHCP and PPPoE subscribers based on an access-line identifier (ALI), you must create a dynamic ALI interface set.

To configure an ALI interface set in a dynamic profile:

1. Access the dynamic profile that defines the ALI interface set.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Configure the dynamic ALI interface set.

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

   Use the predefined variable `$junos-interface-set-name` to represent the name of the ALI interface set. It is replaced with the actual ALI interface set name generated by the router when the first subscriber from that household logs in.

3. Include the underlying interfaces for the dynamic ALI interface set.

```
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”]
user@host# set interface $junos-interface-ifd-name
```

   Use the predefined variable `$junos-interface-ifd-name` to represent the name of the interface. The variable is replaced with the name of the interface on which the subscriber accesses the BNG.

   The `unit` statement is not required in the dynamic profile when you configure an ALI interface set.

4. (Optional) For dynamic PPPoE subscriber interfaces, configure the maximum number of dynamic PPPoE sessions that the router can activate for the ALI interface set; that is, for the same household.

```
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”]
user@host# edit pppoe-underlying-options
[edit dynamic-profiles profile-name interfaces interface-set “$junos-interface-set-name”
pppoe-underlying-options]
user@host# set max-sessions number
```

5. (Optional) Apply attributes for CoS and interface filters to all subscriber interfaces belonging to the ALI interface set.

The following example shows the minimum dynamic profile required to define an ALI interface set named `ali-vlan-set-profile`. It uses predefined variables to represent the interface set and the underlying physical interface.

```
[edit dynamic-profiles ali-vlan-set-profile]
interfaces {
```

```
    interface-set "$junos-interface-set-name" {
        interface "$junos-interface-ifd-name";
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers | 66
Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers | 76
Clearing Access-Line-Identifier Interface Sets | 78
Applying CoS Attributes to VLANs Using Access-Line Identifiers
Access-Line-Identifier-Based Dynamic VLANs Overview | 61

Configuring Dynamic Underlying VLAN Interfaces to Use Access-Line Identifiers

After you define the access-line-identifier (ALI) interface set, you must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on the ALI. You can configure the underlying VLAN interface statically or dynamically.

This topic describes how to configure the underlying VLAN interface *dynamically*.

Before you begin:

- Create a dynamic profile that defines the underlying VLAN interface.

  See the following topics:

  - *Configuring a Basic Dynamic Profile*
  - "Configuring a Dynamic Profile Used to Create Single-Tag VLANs" on page 15
  - "Configuring a Dynamic Profile Used to Create Stacked VLANs" on page 19

To configure a dynamic underlying VLAN interface to use the ALI:

1. In the dynamic profile for the underlying VLAN interface, associate the underlying VLAN interface with the line identity dynamic profile that defines the ALI interface set.

```
[edit dynamic-profiles profile-name]
user@host# set interfaces interface-name unit logical-unit-number auto-configure line-
identity dynamic-profile ali-interface-set-profile-name
```

   For example, the following statement in a dynamic profile named ali-vlan-underlying-profile-demux associates the dynamic underlying VLAN interface with the dynamic profile ali-vlan-set-profile2 that defines the ALI interface set. You must use the predefined dynamic variable $junos-interface-ifd-name to represent the interface name, and $junos-interface-unit to represent the logical unit number.

```
[edit dynamic-profiles ali-vlan-underlying-profile-demux]
user@host# set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit” auto-
configure line-identity dynamic-profile ali-vlan-set-profile2
```

2. Configure one or more trusted options—the access-line-identifier information—that are accepted to trigger the creation of the dynamic VLAN.

```
[edit dynamic-profiles profile-name]
user@host# set interfaces interface-name unit logical-unit-number auto-configure line-
identity include trusted-option
```

   For example, the following statement specifies that only the ARI is accepted to trigger creation of the VLAN. When the ARI is not received, no VLAN is created.

```
[edit dynamic-profiles ali-vlan-underlying-profile-demux]
user@host# set interfaces “$junos-interface-ifd-name” unit “$junos-interface-unit” auto-
configure line-identity include remote-id
```

The following example shows the dynamic configuration that uses these statements. This configuration enables the underlying dynamic IP demultiplexing (IP demux) VLAN interface to create dynamic subscriber interfaces based on the ARI by applying a single default ALI interface set dynamic profile (ali-vlan-set-profile2) to all households on the VLAN interface.

```
[edit dynamic-profiles ali-vlan-underlying-profile-demux]
interfaces {
    “$junos-interface-ifd-name” {
        unit "$junos-interface-unit" {
```

```
            auto-configure {
                line-identity {
                    dynamic-profile ali-vlan-set-profile2;
                    include {
                        remote-id;
                    }
                }
            }
            vlan-id "$junos-vlan-id";
            demux-options {
                underlying-interface “$junos-interface-ifd-name”;
            }
            family inet {
                unnumbered-address lo0.0 preferred-source-address 198.51.100.20;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers | 66

Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers | 76

Access-Line-Identifier-Based Dynamic VLANs Overview | 61

Configuring Static Underlying VLAN Interfaces to Use Access-Line Identifiers

After you define the access-line-identifier (ALI) interface set, you must configure the underlying VLAN interface to enable creation of dynamic VLAN subscriber interfaces based on the ALI. You can configure the underlying VLAN interface statically or dynamically.

This topic describes how to configure the underlying VLAN interface *statically*.

To configure a static underlying VLAN interface to use the ALI:

1. Associate the static underlying VLAN interface with the line identity dynamic profile that defines the ALI interface set.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number auto-configure line-
identity dynamic-profile ali-interface-set-profile-name
```

For example, the following statement associates static Gigabit Ethernet VLAN interface ge-1/0/0.0 with the dynamic profile ali-vlan-set-profile that defines the ALI interface set.

```
[edit]
user@host# set interfaces ge-1/0/0 unit 0 auto-configure line-identity dynamic-profile ali-
vlan-set-profile
```

2. Configure one or more trusted options—the access-line-identifier information—that are accepted to trigger the creation of the dynamic VLAN.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number auto-configure line-
identity include trusted-option
```

For example, the following statement specifies that only the ARI is accepted to trigger creation of the VLAN. When the ARI is not received, no VLAN is created.

```
[edit]
user@host# set interfaces ge-1/0/0 unit 0 auto-configure line-identity include remote-id
```

The following example shows the static configuration that uses this statement. This configuration enables the underlying VLAN interface ge-1/0/0.0 to create dynamic subscriber interfaces based on the ARI by applying a single default ALI interface set dynamic profile (ali-vlan-set-profile) to all households on the VLAN interface.

```
[edit]
interfaces {
    ge-1/0/0 {
        flexible-vlan-tagging;
        unit 0 {
            vlan-id 100;
            auto-configure {
```

```
                line-identity {
                    dynamic-profile ali-vlan-set-profile;
                    include {
                        remote-id;
                    }
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers | 66
Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers | 76
Access-Line-Identifier-Based Dynamic VLANs Overview | 61

Configuring Dynamic VLAN Subscriber Interfaces Based on Access-Line Identifiers

After you define the dynamic access-line-identifier (ALI) interface set and enable creation of ALI-based dynamic VLAN subscriber interfaces on the underlying VLAN interface, you must complete the configuration by associating the ALI interface set with the PPPoE or IP demultiplexing (IP demux) subscriber interface in the dynamic profile for the subscriber interface.

Before you begin:

- Create a dynamic profile that defines the logical subscriber interface.

  See the following topics:

  - *Configuring a Basic Dynamic Profile*
  - "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194

To configure a dynamic VLAN subscriber interface based on the ALI:

- In the dynamic profile for the PPPoE or IP demux subscriber interface, associate the dynamic ALI interface set with the dynamic VLAN subscriber interface name (`pp0` or `demux0`) and logical unit number.

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

  For example, the following statement in a dynamic profile named ali-vlan-pppoe-profile associates the dynamic ALI interface set with the dynamic pp0 (PPPoE) logical subscriber interface. You must use the predefined dynamic variable $junos-interface-set-name to represent the name of the dynamic ALI interface set, and $junos-interface-unit to represent the logical unit number of the subscriber interface.

  ```
  [edit dynamic-profiles ali-vlan-pppoe-profile]
  user@host# set interfaces interface-set $junos-interface-set-name interface pp0 unit $junos-
  interface-unit
  ```

  Similarly, the following statement in a dynamic profile named ali-vlan-demux-profile associates the dynamic ALI interface set (represented by $junos-interface-set-name) with the demux0 (IP demux) logical subscriber interface.

  ```
  [edit dynamic-profiles ali-vlan-demux-profile]
  user@host# set interfaces interface-set $junos-interface-set-name interface demux0 unit
  $junos-interface-unit
  ```

The following examples show the dynamic configurations that use each of these statements. The following sample configuration shows a dynamic profile named ali-vlan-pppoe-profile for an ALI-based dynamic PPPoE (pp0) subscriber interface for use by PPPoE subscribers.

```
[edit dynamic-profiles ali-vlan-pppoe-profile]
interfaces {
    interface-set "$junos-interface-set-name" {
        interface pp0 {
            unit "$junos-interface-unit";
        }
    }
    pp0 {
        unit "$junos-interface-unit" {
            ppp-options {
                chap;
```

```
                pap;
            }
            pppoe-options {
                underlying-interface "$junos-underlying-interface";
                server;
            }
            no-keepalives;
            family inet {
                unnumbered-address lo0.0;
            }
        }
    }
}
```

The following sample configuration shows a dynamic profile named ali-vlan-demux-profile for an ALI-based dynamic IP demux (demux0) subscriber interface for use by DHCP subscribers.

```
[edit dynamic-profiles ali-vlan-demux-profile]
interfaces {
    interface-set "$junos-interface-set-name" {
        interface demux0 {
            unit "$junos-interface-unit";
        }
    }
    demux0 {
        unit "$junos-interface-unit" {
            demux-options {
                underlying-interface "$junos-underlying-interface";
            }
            family inet {
                demux-source {
                    $junos-subscriber-ip-address;
                }
                unnumbered-address lo0.0 preferred-source-address  198.51.100.202;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers

IN THIS SECTION

Purpose

View information about dynamic access-line-identifier (ALI) interface sets and ALI-based dynamic VLAN subscriber interfaces configured on the router.

Action

- To display the logical and physical interface associations for the classifier, rewrite rules, scheduler map objects, and CoS adjustment settings:

```
user@host> show class-of-service interface interface-name
```

- To display the CoS associations for the specified dynamic ALI interface set:

```
user@host> show class-of-service interface-set ali-interface-set-name
```

- To display information about the specified CoS traffic shaping and scheduling profile:

```
user@host> show class-of-service traffic-control-profile profile-name
```

- To display address bindings and ALI interface set information in the client table on the extended DHCP local server:

```
user@host> show dhcp server binding detail
```

- To display status information about a specified Gigabit Ethernet interface:

```
user@host> show interfaces ge-fpc/pic/port.logical-unit-number
```

- To display status information about a specified IP demultiplexing (IP demux) interface:

```
user@host> show interfaces demux0.logical-interface-number
```

- To display information about all dynamic ALI interface sets configured on the router:

```
user@host> show interfaces interface-set
```

- To display session-specific information about ALI-based dynamic PPPoE subscriber interfaces:

```
user@host> show pppoe interfaces pp0.logical-unit-number
```

- To display information about PPPoE underlying interfaces, including whether creation of ALI-based dynamic VLAN subscriber interfaces is enabled on the underlying interface:

```
user@host> show pppoe underlying-interfaces logical-interface-name detail
```

- To display information about active subscriber sessions associated with ALI interface sets:

```
user@host> show subscribers detail
```

- To display information about active subscriber sessions associated with a specified ALI interface set:

```
user@host> show subscribers ali-interface-set-name ali-interface-set-name detail
```

- To display information about active subscriber sessions that have an access-line-identifier value containing a matching substring:

```
user@host> show subscribers agent-remote-identifier agent-remote-identifier-substring detail
```

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers | 66
Clearing Access-Line-Identifier Interface Sets | 78
Access-Line-Identifier-Based Dynamic VLANs Overview | 61

Clearing Access-Line-Identifier Interface Sets

IN THIS SECTION

Purpose

Clear a specified dynamic access-line-identifier (ALI) interface set configured on the router.

Action

- To clear a specified ALI interface set that has no active members:

```
user@host> clear auto-configuration interfaces interface-set interface-set-name
```

For example, the following command clears the ALI interface set named ari-1003-ge-1/0/0.4001:

```
user@host> clear auto-configuration interfaces interface-set ari-1003-ge-1/0/0.4001
Interface-set ari-1003-ge-1/0/0.4001 deleted
```

Meaning

When configured to do so, the router dynamically creates an ALI interface set when the first DHCP or PPPoE subscriber from a particular household logs in. However, the router does not automatically delete the ALI interface set when the last subscriber from that household logs out. As a result, you must use the `clear auto-configuration interfaces interface-set` command to explicitly clear the ALI interface set when it no longer has any active subscriber interface members. If you attempt to clear an ALI interface that still has active member interfaces, the router displays an error message and rejects the command.

When you specify the name of the ALI interface set to be cleared, you must use the ALI interface set name internally generated by the router, and not the actual ALI string carried in DHCP and PPPoE control packets. The router uses the following format to name ALI interface sets:

trusted-option-nnnn-interface-name.logical-unit-number

where:

- *trusted-option* is a prefix identifying the access-line identifier that was configured to be accepted and which triggered creation of the interface set:
  - `aci`—The trusted option is the ACI.
  - `ari`—The trusted option is the ARI.
  - `aci+ari`—Both the ACI and the ARI are trusted options and both were received.
  - `noids`—Neither the ACI nor the ARI is configured as the trusted option and neither ACI nor ARI is received.
- *nnnn* is a randomly generated 4-digit identifier; for example, 1003.
- *interface-name* is the name of the dynamic subscriber interface; for example, ge-1/0/0 or demux0.

- *logical-unit-number* is the logical unit number of the dynamic subscriber interface; for example, 4001.

The following are all examples of generated interface set names:

```
aci-1003-ge-1/0/0.4001
ari-4297-demux0.3221225524
aci+ari-8115-demux0.4255221223
noids-3232-ge-2/1/0.1234
```

To view the names of the ALI interface sets configured on the router, use the `show subscribers` command.

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers | **66**

Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers | **76**

CLI Explorer

CHAPTER 6

High Availability for Service VLANs

IN THIS CHAPTER

Ethernet OAM Support for Service VLANs Overview

IN THIS SECTION

You can enable propagation of the Ethernet IEEE 802.1ag Operation, Administration, and Maintenance (OAM) state of a static single-tagged service VLAN (S-VLAN) to a dynamic or static double-tagged customer VLAN (C-VLAN) and, by extension, to the subscriber interfaces configured on the C-VLAN. The static S-VLAN *logical interface* must be configured on a Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet physical interface.

Propagation of the S-VLAN OAM state to associated C-VLANs ensures that when the OAM state of the S-VLAN link is down, the associated C-VLANs and all subscriber interfaces configured on the C-VLANs are brought down as well.

Ethernet OAM Support for Service VLANs Terms and Acronyms

Table 5 on page 82 defines the basic terms and acronyms used in this discussion of Ethernet OAM support for service VLANs.

Table 5: Ethernet OAM Support for Service VLANs Terms and Acronyms

Term	Definition
CFM	Connectivity fault management. Provides end-to-end monitoring of an Ethernet network that can be made up of one or more service instances. Junos OS supports Ethernet IEEE 802.1ag CFM.
Continuity check protocol	A feature of Ethernet IEEE 802.1ag CFM that provides fault detection within a maintenance association.
C-VLAN	Customer VLAN. A dynamic or static double-tagged logical interface that has both an outer VLAN tag (corresponding to the S-VLAN) and an inner VLAN tag (corresponding to the C-VLAN). In a 1:1 subscriber network access model, dedicated C-VLANs provide a one-to-one correspondence between an individual subscriber and the VLAN encapsulation.
OAM	Operation, Administration, and Maintenance. A set of Ethernet connectivity specifications and functions providing connectivity monitoring, fault detection and notification, fault verification, fault isolation, loopback, and remote defect identification. Ethernet interfaces on MX Series routers support the IEEE 802.1ag standard for OAM.
S-VLAN	Service VLAN. A static single-tagged logical interface that has only one outer VLAN tag (corresponding to the S-VLAN). In an N:1 subscriber network access model, S-VLANs are dedicated to a particular service, such as video, voice, or data, instead of to a particular subscriber. Because an S-VLAN is typically shared by many subscribers within the same household or in different households, it provides a many-to-one correspondence between individual subscribers and the VLAN encapsulation.
VLAN	Virtual local area network. A logical group of network devices that appear to be on the same local area network, regardless of their physical location.

Components of Ethernet OAM Support for Service VLANs

Ethernet OAM support for S-VLANs involves the following components:

- Physical interface—On MX Series routers with Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces, you can enable propagation of the S-VLAN OAM state to a C-VLAN on Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet physical interfaces.

- S-VLAN—To enable propagation of the S-VLAN Ethernet OAM state to associated C-VLANs and subscriber interfaces, configure the static single-tagged S-VLAN logical interface to run the Ethernet IEEE 802.1ag CFM continuity check protocol.
- C-VLAN—The C-VLAN is a dynamic or static double-tagged logical interface that has the same S-VLAN (outer) tag as the static single-tagged S-VLAN logical interface. If propagation of the S-VLAN OAM state to the C-VLAN is enabled on the physical interface, the router brings down the C-VLAN and its associated subscriber interfaces when the CFM continuity check protocol detects that the OAM state of the underlying S-VLAN is down.
- Subscriber interfaces—Propagation of the S-VLAN Ethernet OAM state to associated C-VLANs and subscriber interfaces applies to all dynamic or static DHCP, IP demultiplexing (IP demux), and PPPoE subscriber interfaces configured on the C-VLAN.

How Ethernet OAM Support for Service VLANs Works

To enable propagation of the Ethernet OAM state of the S-VLAN to associated C-VLANs and subscriber interfaces, use the `oam-on-svlan` statement when you configure a Gigabit Ethernet (ge), 10-Gigabit Ethernet (xe), or aggregated Ethernet (ae) physical interface.

If Ethernet IEEE 802.1ag CFM is properly configured on the S-VLAN logical interface, including the `oam-on-svlan` statement for these Ethernet interfaces causes the router to bring down both of the following when the CFM continuity check protocol detects that the OAM state of the S-VLAN logical interface is down:

- All dynamic or static double-tagged C-VLAN logical interfaces that have the same S-VLAN (outer) tag as the S-VLAN logical interface on which they are configured.
- All dynamic or static DHCP, IP demux, and PPPoE logical subscriber interfaces configured on the associated C-VLANs.

To illustrate how Ethernet OAM support for S-VLANs works, consider the following sample configuration on a Gigabit Ethernet physical interface:

- Gigabit Ethernet physical interface ge-1/0/3 configured with the `svlan-on-oam` statement.
- Static single-tagged S-VLAN logical interface ge-1/0/3.0, which has a single S-VLAN outer tag, VLAN ID 600.
- Ethernet OAM CFM protocol configured on the static S-VLAN logical interface. The CFM configuration includes an action profile with the `interface-down` default action to bring down the C-VLAN and dynamic subscriber interfaces when the continuity check protocol detects that the Ethernet OAM state of S-VLAN interface ge-1/0/3.0 is down.
- Static double-tagged C-VLAN logical interface ge-1/0/3.100, which has an S-VLAN outer tag, VLAN ID 600, and a C-VLAN inner tag, VLAN ID 1.

- Static PPPoE subscriber interfaces configured on C-VLAN interface ge-1/0/3.100.

Because the S-VLAN and C-VLAN logical interfaces in this example have the same S-VLAN outer tag (VLAN ID 600), the router brings down the C-VLAN interface and the PPPoE logical subscriber interfaces when the CFM continuity check detects that the OAM status of S-VLAN interface ge-1/0/3.0 is down.

Restrictions for Using Ethernet OAM Support for Service VLANs

Ethernet OAM support for S-VLANs is *not currently supported* for use with any of the following:

- Dynamically configured S-VLAN logical interfaces
- S-VLAN trunk interfaces
- C-VLAN trunk interfaces

RELATED DOCUMENTATION

Configuring Ethernet OAM Support for Service VLANs with Double-Tagged Customer VLANs | 84
IEEE 802.1ag OAM Connectivity Fault Management Overview

Configuring Ethernet OAM Support for Service VLANs with Double-Tagged Customer VLANs

You can enable propagation of the Ethernet IEEE 802.1ag Operation, Administration, and Maintenance (OAM) state of a static single-tagged service VLAN (S-VLAN) to the dynamic or static double-tagged customer VLAN (C-VLAN) that has the same S-VLAN (outer) tag as the S-VLAN, and, by extension, to subscriber interfaces configured on the C-VLAN. The static S-VLAN logical interface must be configured on a Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet physical interface.

Before you begin:

- Make sure the static single-tagged S-VLAN logical interface is configured with the Ethernet 802.1ag OAM connectivity fault management (CFM) continuity check protocol.

  See IEEE 802.1ag OAM Connectivity Fault Management Overview.

To enable propagation of the Ethernet OAM state of a static single-tagged S-VLAN to dynamic or static double-tagged C-VLAN logical interfaces:

- Configure a Gigabit Ethernet (ge), 10-Gigabit Ethernet (xe), or aggregated Ethernet (ae) physical interface to propagate the S-VLAN Ethernet OAM state to C-VLAN logical interfaces that have the same S-VLAN (outer) tag as the S-VLAN interface.

```
[edit]
user@host# set interfaces interface-name-fpc/pic/port oam-on-svlan
```

For example, the following statement enables propagation of the Ethernet OAM state of a static single-tagged S-VLAN on Gigabit Ethernet interface ge-1/0/5 to a dynamic or static double-tagged C-VLAN logical interface with the same S-VLAN (outer) tag as the S-VLAN interface.

```
[edit]
user@host# set interfaces ge-1/0/5 oam-on-svlan
```

Including the `oam-on-svlan` statement when you configure a Gigabit Ethernet, 10-Gigabit Ethernet, or aggregated Ethernet physical interface causes the router to bring down both of the following when the CFM continuity check protocol detects that the OAM state of the S-VLAN logical interface is down:

- All dynamic or static double-tagged C-VLANs on the S-VLAN interface that have the same S-VLAN (outer) tag as the S-VLAN interface.
- All DHCP, IP demultiplexing (IP demux), and PPPoE logical subscriber interfaces configured on the associated C-VLANs.

Example: Gigabit Ethernet Interface with Static S-VLAN, Dynamic C-VLAN, and Dynamic PPPoE Subscriber Interfaces

The following example shows a dynamic subscriber access configuration that uses the `oam-on-svlan` statement on a Gigabit Ethernet interface. This example configures Gigabit Ethernet physical interface ge-1/0/5 with a static single-tagged S-VLAN logical interface (ge-1/0/5.1) that runs the Ethernet 802.1ag OAM CFM continuity check protocol. A dynamic profile named double-vlans creates a dynamic double-tagged C-VLAN interface, and a dynamic profile named pppoe-profile creates dynamic PPPoE subscriber interfaces on the C-VLAN interface. The `oam-on-svlan` statement for ge-1/0/5 propagates the Ethernet OAM state of S-VLAN interface ge-1/0/5.1 to the C-VLAN interface and the dynamic PPPoE subscriber interfaces.

For clarity, the configuration is divided into five steps.

1. Configure a dynamic profile named double-vlans that defines a dynamic double-tagged C-VLAN logical interface.

```
[edit]
dynamic-profiles {
    double-vlans {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-interface-unit" {
                    vlan-tags outer "$junos-stacked-vlan-id" inner "$junos-vlan-id";
                    encapsulation ppp-over-ether;
                    pppoe-underlying-options {
                        dynamic-profile pppoe-profile;
                    }
                }
            }
        }
    }
}
```

2. Configure a dynamic profile named pppoe-profile that defines dynamic PPPoE subscriber interfaces on the C-VLAN.

```
[edit]
dynamic-profiles {
    pppoe-profile {
        interfaces {
            pp0 {
                unit "$junos-interface-unit" {
                    pppoe-options {
                        underlying-interface "$junos-underlying-interface";
                        server;
                    }
                    family inet {
                        unnumbered-address lo0.0;
                    }
                }
            }
        }
    }
}
```

3. Configure Gigabit Ethernet physical interface ge-1/0/5.

```
[edit]
interfaces {
    ge-1/0/5 {
        description "connect to remote router";
        flexible-vlan-tagging;
        oam-on-svlan;
        unit 1 {
            vlan-id 1;
        }
        auto-configure {
            stacked-vlan-ranges {
                dynamic-profile double-vlans {
                    accept any;
                    ranges {
                        any,any;
                    }
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 198.51.1.1/32 {
                    primary;
                }
            }
        }
    }
}
```

The preceding example in Step 3 configures a static, single-tagged S-VLAN logical interface (ge-1/0/5.1) with VLAN ID 1, and references the double-vlans dynamic profile to create a dynamic double-tagged C-VLAN logical interface with S-VLAN (outer) tag `any` and C-VLAN (inner) tag `any`. The tag value `any` represents the entire range of VLAN IDs or S-VLAN IDs, including VLAN ID 1.

Because the C-VLAN outer tag (`any`) matches the S-VLAN tag VLAN ID 1, the `oam-on-svlan` statement in the configuration causes the router to propagate the Ethernet OAM state of S-VLAN ge-1/0/5.1 to the dynamic double-tagged C-VLAN logical interface (created by the double-vlans dynamic profile) and, by extension, to the dynamic PPPoE subscriber interfaces on the C-VLAN (created by the pppoe-profile dynamic profile).

4. Configure the Ethernet 802.1ag OAM CFM continuity check protocol on the static S-VLAN interface (ge-1/0/5.1).

```
[edit]
protocols {
    oam {
        ethernet {
            connectivity-fault-management {
                action-profile myDefault {
                    default-actions {
                        interface-down;
                    }
                }
                maintenance-domain md1 {
                    level 1;
                    maintenance-association ma1 {
                        continuity-check {
                            interval 1s;
                        }
                        mep 100 {
                            interface ge-1/0/5.1;
                            direction down;
                            remote-mep 101 {
                                action-profile myDefault;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

If the CFM continuity check protocol detects that the Ethernet OAM state of S-VLAN interface ge-1/0/5.1 is down, the `interface-down` action in the myDefault action profile causes the router to bring down both of the following:

- The dynamic double-tagged C-VLAN logical interface that has the same S-VLAN (outer) tag as S-VLAN interface ge-1/0/5.1

- The dynamic PPPoE subscriber interfaces configured on the dynamic C-VLAN interface

5. Create a PPP access profile.

   For brevity, this configuration is only partially shown. The missing portions of the configuration are replaced with ellipses (...).

```
[edit]
access {
    ...
    profile ppp-authenticator {
        ...
    }
}
```

RELATED DOCUMENTATION

Ethernet OAM Support for Service VLANs Overview | **81**

IEEE 802.1ag OAM Connectivity Fault Management Overview

2

PART

Configuring DHCP Subscriber Interfaces

CHAPTER 7

VLAN and Demux Subscriber Interfaces Overview

IN THIS CHAPTER

DHCP Subscriber Interface Overview

IN THIS SECTION

You can identify subscribers statically or dynamically.

To identify subscribers statically, you can reference a static VLAN interface in a dynamic profile. To identify subscribers dynamically, you create variables for demux interfaces that are dynamically created by DHCP when subscribers log in.

Statically Identifying Subscribers

Before you can configure static subscriber interfaces in a dynamic profile, you must first configure the logical interfaces on the router to which you expect clients to connect. After you have created the static interfaces, you can modify them by using dynamic profiles to apply configuration parameters.

You can also configure subscribers by creating sets of static IP demux interfaces that are not referenced in a dynamic profile.

When configuring the interfaces stanza within a dynamic profile, you use variables to specify the interface name and the logical unit value. When a DHCP subscriber sends a DHCP request to the interface, the dynamic profile replaces the `interface-name` and `unit` variables with the actual interface name and logical unit number of the interface that received the DHCP request. After this association is made, the router configures the interface with any CoS or protocol (that is, IGMP) configuration within the dynamic profile, or applies any input or output filter configuration that you have associated with that dynamic profile.

```
[edit dynamic-profiles]
interfaces interface-name {
    unit logical-unit-number {
        family family {
            address address;
            filter {
                input filter-name;
                output filter-name;
            }
        unnumbered-address interface-name <preferred-source-address address>;
        vlan-id;
    }
    vlan-tagging;
}
```

Dynamically Identifying Subscribers

You can configure demux interfaces to represent a subscriber interface in a dynamic profile. When a subscriber logs in using a DHCP access method, the demux interface is dynamically created.

You specify variables for the unit number, the name of the underlying interface, and the IP address in the dynamic profile. These variables are replaced with the values that are supplied by DHCP when the subscriber logs in.

RELATED DOCUMENTATION

Static Subscriber Interfaces and VLAN Overview | 7

Subscriber Interfaces and Demultiplexing Overview | 93

Subscriber Interfaces and Demultiplexing Overview

IN THIS SECTION

You can create logical subscriber interfaces using static or dynamic demultiplexing interfaces. In addition, you can use either IP demultiplexing interfaces or VLAN demultiplexing interfaces when creating logical subscriber interfaces.

Demultiplexing (demux) interfaces are logical interfaces that share a common, underlying *logical interface* (in the case of IP demux) or underlying physical interface (in the case of VLAN demux). You can use these interfaces to identify specific subscribers or to separate individual circuits by IP address (IP demux) or VLAN ID (VLAN demux).

The subscriber interfaces can provide different levels of services for individual subscribers in an access network. For example, you can apply CoS parameters for each subscriber.

From Junos OS Release 18.1 onwards, packet triggered subscribers feature creates IP demultiplexing interfaces (IP demux IFL) on receiving a data packet from clients with pre-assigned IP address. The IP demultiplexing interfaces are created for both IPv4 or IPv6 data packets. On receiving the packets, the forwarding plane checks the source IP address. If the source IP address matches any one of the configured IP address or prefix ranges, the subscriber is sent to the Routing engine. The Routing Engine authenticates the subscriber with authenticating server. The authenticating server requests for volume accounting and may also request for advanced services such as firewall filter or CoS. The IP demux IFL is created with the services requested by the authenticating server. The IP demux IFL employs subscriber services in networks with statically assigned IP clients or subscribers with pre-assigned IP address.

If the source IP address does not fall within any of the IP address or prefix ranges on the interface, the IP demux IFL does not get created

Interface Sets of Static Demux Interfaces

You can group static demux interfaces to create individual subscriber interfaces using interface sets. Interface sets enable you to provide the same level of service for a group of subscribers; for example, all residential subscribers who receive the basic data service.

Figure 4 on page 94 shows a subscriber interface configured using a set of IP demux interfaces with an underlying VLAN interface.

Figure 4: IP Demux Subscriber Interface

Dynamic Demultiplexing Interfaces

You can configure demux interfaces to represent a dynamic subscriber interface in a dynamic profile.

Demux interfaces are dynamically created by a DHCP access method when the underlying interface for the demux interface is configured for the access method. The DHCP access model creates the demux interface with the subscriber's assigned IP address (for IP demux interfaces) or VLAN ID (for VLAN demux interfaces).

To configure an IP demux interface in the dynamic profile, you specify variables for the unit number, the name of the underlying interface, and the IP address. To configure a VLAN demux interface in the dynamic profile, you specify variables for the unit number, the name of the underlying interface, and the VLAN ID. These variables are replaced with the values that are supplied by DHCP when the subscriber logs in.

Guidelines for Configuring Demux Interfaces for Subscriber Access

When you configure static or dynamic demux interfaces for subscriber access, consider the following guidelines:

- Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.
- You can configure only one `demux0` interface per chassis.
- For IP demux interfaces, you can define logical demux interfaces on top of the `demux0` interface (for example, `demux0.1`, `demux0.2`, and so on).
- Hierarchical and per-unit scheduling is supported for dynamically created demux interfaces on the EQ DPC.
- IP demux interfaces support IPv4 (`family inet`) and IPv6 (`family inet6`).

- IP demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has other cards in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.
- You can configure IPv4 and IPv6 addressing for static and dynamic demux interfaces.
- Demux interfaces currently support only Gigabit Ethernet, Fast Ethernet, 10-Gigabit Ethernet, and aggregated Ethernet underlying interfaces.
- You must associate IP demux interfaces with an underlying logical interface.
- You must associate VLAN demux interfaces with an underlying device (physical interface).
- You cannot use a dynamic demux interface to represent multiple subscribers in a dynamic profile attached to an interface. One dynamic demux interface represents one subscriber. Do not configure the `aggregate-clients` option when attaching a dynamic profile to a demux interface for DHCP.

> **CAUTION**: Before you make any changes to the underlying interface for a demux0 interface, you must ensure that no subscribers are currently present on that underlying interface. If any subscribers are present, you must remove them before you make changes.

RELATED DOCUMENTATION

Configuring a Subscriber Interface Using a Set of Static IP Demux Interfaces | 97
Configuring a Subscriber Interface Using a Set of Static VLAN Demux Interfaces | 99
Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles
Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles | 101
Demultiplexing Interface Overview

IP Demux Interfaces over Static or Dynamic VLAN Demux Interfaces

You can configure a router with IP demux interfaces over VLAN demux interfaces. Just as IP demux interfaces demultiplex their underlying VLAN demux interfaces based on IP address, VLAN demux interfaces demultiplex their underlying aggregate Ethernet or Ethernet interfaces based on VLAN ID.

When configuring IP demux interfaces over VLAN demux interfaces, keep the following in mind:

- Only single and dual VLAN tag options are supported as VLAN selectors.

- Both inet and inet6 families are supported.
- All firewall and CoS features are supported.
- Both static and dynamic VLAN demux interface creation is supported.
- Only MPCs are supported.

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | **93**

Distribution of Demux Subscribers in an Aggregated Ethernet Interface

Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet | **120**

Example: Dynamic IP Demux Subscriber Interfaces over Dynamic VLAN Demux Interfaces | **103**

Example: Concurrent Configuration of Dynamic DHCP IP Demux and PPPoE Demux Interfaces over the Same VLAN Demux Interface | **151**

Aggregated Ethernet Interfaces Overview

CHAPTER 8

Configuring Sets of Demux Interfaces to Provide Services to a Group of Subscribers

IN THIS CHAPTER

Configuring a Subscriber Interface Using a Set of Static IP Demux Interfaces

You can create logical subscriber interfaces from IP demux interfaces. IP demultiplexing (demux) interfaces are logical interfaces that share a common, underlying logical interface. IP demux interfaces can be used to identify specific subscribers or to separate individual circuits.

You can group individual subscriber interfaces using interface sets to provide the same level of service for a group of subscribers; for example, all residential subscribers who receive the basic data service. Interface sets can be defined as a list of logical interfaces (unit 0, unit 1, and so on).

> **NOTE**: Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.

To configure a group of static IP demux interfaces:

1. Configure the interface set.

```
interfaces {
    interface-set demux-set {
        interface demux0 {
            unit 0;
            unit 1;
            }
```

```
        }
    }
```

2. Define the units of the interface set.

```
demux0 {
    unit 0 {
        demux-options {
            underlying-interface ge-2/0/1.1;
        }
        family inet {
            demux-source {
                 203.0.113.0/24;
            }
            address 203.0.113.25/24;
        }
    }
    unit 1 {
        demux-options {
            underlying-interface ge-2/0/1.1;
        }
        family inet {
            demux-source {
                203.0.133.110/24;
            }
            address 203.0.113.12/24;
        }
    }
}
```

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

Configuring a Subscriber Interface Using a Set of Static VLAN Demux Interfaces

You can create logical subscriber interfaces from VLAN demux interfaces. VLAN demultiplexing (demux) interfaces are logical interfaces that share a common, underlying physical interface. VLAN demux interfaces can be used to identify specific subscribers or to separate individual circuits.

You can group individual subscriber interfaces using interface sets to provide the same level of service for a group of subscribers; for example, all residential subscribers who receive the basic data service. Interface sets can be defined as a list of logical interfaces (unit 0, unit 1, and so on).

> **NOTE**: Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.

To configure a group of static VLAN demux interfaces:

1. Configure the interface set.

```
interfaces {
    interface-set demux-set {
        interface demux0 {
            unit 0;
            unit 1;
            }
        }
    }
```

2. Define the units of the interface set.

```
demux0 {
    unit 0 {
        vlan-id 10;
        demux-options {
            underlying-interface ge-2/0/1;
        }
        family inet {
            address 203.0.113.201/24;
        }
    }
    unit 1 {
```

```
        vlan-id 20;
        demux-options {
            underlying-interface ge-2/0/1;
        }
        family inet {
            address 203.0.113.202/24;
        }
    }
}
```

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

CHAPTER 9

Configuring Dynamic Demux Interfaces That are Created by DHCP

IN THIS CHAPTER

Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles

You can configure dynamic subscriber interfaces using VLAN demux interfaces.

To enable the dynamic demux interface to be created by DHCP, you configure the demux options in a dynamic profile. Dynamic profiles enable you to dynamically apply configured values (including CoS, IGMP, or filter configuration) to the dynamic interfaces, making them easier to manage.

NOTE: Only demux0 is supported. If you configure another demux interface, such as demux1, the configuration commit fails.

Before you begin:

- Configure the dynamic profile.

  See *Configuring a Basic Dynamic Profile*.

To configure dynamic subscriber interfaces:

1. Specify that you want to configure the `demux0` interface in the dynamic profile.

```
user@host#  edit dynamic-profiles business-profile interfaces demux0
```

2. Configure the unit for the `demux0` interface.

a. Configure the variable for the unit number of the `demux0` interface.

The variable is dynamically replaced with the unit number that DHCP supplies when the subscriber logs in.

```
[edit dynamic-profiles business-profile interfaces demux0]
user@host# edit unit $junos-interface-unit
```

b. Configure the variable for the underlying interface of the demux interfaces by specifying the `$junos-interface-ifd-name` variable.

The variable is dynamically replaced with the underlying device name that DHCP supplies when the subscriber logs in.

```
[edit dynamic-profiles business-profile interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-interface-ifd-name
```

c. Configure the variable for the VLAN ID.

```
[edit dynamic-profiles business-profile interfaces demux0 unit “$junos-interface-unit”]
user@host# set vlan-id $junos-vlan-id
```

3. Configure the family for the demux interfaces.

a. Specify that you want to configure the family.

For IPv4:

```
[edit dynamic-profiles business-profile interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

For IPv6:

```
[edit dynamic-profiles business-profile interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet6
```

b. Configure the unnumbered address for the family.

```
[edit dynamic-profiles business-profile interfaces demux0 unit “$junos-interface-unit”
family inet]
user@host#  set unnumbered-address lo0.0
```

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93
Configuring MAC Address Validation for Subscriber Interfaces | 173
Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148
Example: Dynamic IP Demux Subscriber Interfaces over Dynamic VLAN Demux Interfaces | 103

Example: Dynamic IP Demux Subscriber Interfaces over Dynamic VLAN Demux Interfaces

IN THIS SECTION

This example describes how to configure dynamic IP demux interfaces over dynamic VLAN demux interfaces.

Requirements

Before you begin, make sure to configure either DHCP Relay or DHCP Local Server. For information about configuring either of these components, see *Extended DHCP Relay Agent Overview* or *Understanding Differences Between Legacy DHCP and Extended DHCP*.

Also, before you begin, see the conceptual information about VLAN demux interfaces in:

- "Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces" on page 148
- "Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles" on page 101

Overview

You can create a subscriber interface using an IP demux interface stacked on a static or dynamic VLAN demux interface. IP demux interfaces are used to uniquely identify subscribers in an access network based on their IP address.

Configuration

IN THIS SECTION

Preparing a Subscriber Access Interface

CLI Quick Configuration

To quickly configure the aggregated Ethernet interface over which subscribers access the router:

```
[edit]
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-5/0/9 gigether-options 802.3ad ae0
set interfaces ge-5/1/9 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux accept inet
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux ranges 500-1000
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp link-protection
```

Step-by-Step Procedure

You must configure an interface over which clients initially access the router. We recommend that you specify the same VLAN tagging for the interface that you expect from incoming clients. This example uses flexible VLAN tagging to simultaneously support transmission of 802.1Q VLAN single-tag and dual-tag frames on logical interfaces on the same Ethernet port.

If you want it to automatically create dynamic VLANs, the interface must include the VLAN range type (single or stacked) and contain any specific ranges you want the VLANs to use.

To configure an interface for subscriber access:

1. Configure the number of aggregated Ethernet interfaces on the router.

```
[edit]
user@host# set chassis aggregated-devices ethernet device-count 1
```

2. Access the physical interface over which you want subscribers to initially access the router.

```
[edit]
user@host# edit interfaces ge-5/0/9
```

3. Specify the aggregated Ethernet interface to which the physical interface belongs.

```
[edit interfaces ge-5/0/9]
user@host# set gigether-options 802.3ad ae0
```

4. Repeat Step 2 and Step 3 for each interface you want to assign to the aggregated Ethernet bundle.

```
[edit]
user@host# set interfaces ge-5/1/9 gigether-options 802.3ad ae0
```

5. Access the aggregated Ethernet interface.

```
[edit]
user@host# edit interfaces ae0
```

6. Specify the VLAN tagging that you want the aggregated Ethernet interfaces to use.

```
[edit interfaces ae0]
user@host# set vlan-tagging
```

7. Edit the `auto-configure` stanza to automatically configure VLANs.

```
[edit interfaces ae0]
user@host# edit auto-configure
```

8. Edit the `vlan-ranges` stanza for single-tagged VLANs.

```
[edit interfaces ae0 auto-configure]
user@host# edit vlan-ranges
```

9. Specify the dynamic VLAN profile that you want the interface to use for dynamically creating single-tagged VLANs.

```
[edit interfaces ae0 auto-configure vlan-ranges]
user@host# edit dynamic-profile Auto-VLAN-Demux
```

10. Specify what VLAN Ethernet packet type the VLAN profile accepts.

```
[edit interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux]
user@host# set accept inet
```

11. Specify the VLAN ranges that you want the dynamic profile to use. The following example specifies a lower VLAN ID limit of 500 and an upper VLAN ID limit of 1000.

```
[edit interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux]
user@host# set ranges 500-1000
```

12. (Optional) Activate the transmission of LACP packets on the aggregated Ethernet interfaces.

```
[edit interfaces ae0]
user@host# set aggregated-ether-options lacp active
```

13. Specify that the aggregated Ethernet interfaces use link protection.

```
[edit interfaces ae0]
user@host# set aggregated-ether-options lacp link-protection
```

Preparing the Loopback Interface

CLI Quick Configuration

To quickly configure the required loopback interface for this example:

```
[edit]
set interfaces lo0.0 unit 0 family inet address 198.51.100.100/32
```

Step-by-Step Procedure

You must configure a loopback interface for use as the unnumbered address and preferred source address for dynamically created interfaces.

To configure the required loopback interface for this example:

1. Configure a loopback interface.

```
[edit]
user@host# edit interfaces lo0.0
```

2. Specify that the loopback interface accept inet packets.

```
[edit interfaces lo0 unit 0]
user@host# edit family inet
```

3. Specify the IP address for the loopback interface.

```
[edit interfaces lo0 unit 0 family inet]
user@host# set address 198.51.100.100/32
```

Configuring a Dynamic Profile to Dynamically Create Single-Tagged VLANs

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create single-tagged VLANs in the example:

```
[edit]
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux-source
inet
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit proxy-arp
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-id $junos-
vlan-id
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux options
underlying-interface $junos-interface-ifd-name
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit family inet
unnumbered-address lo0.0 preferred source-address 198.51.100.100
```

Step-by-Step Procedure

For dynamic IP demux interfaces to reside on a dynamic VLAN demux interface, the VLAN interface must first exist.

A dynamic profile that configures a VLAN demux interface must specify variables for unit, underlying interface name, and VLAN ID. A dynamic VLAN demux interface associates specific subscribers to separate individual circuits by VLAN ID.

To configure a dynamic profile and attach it to a dynamic VLAN demux interface so that it automatically creates VLAN interfaces:

1. Create a dynamic profile for automatically creating single-tagged VLAN interfaces.

```
[edit]
user@host# edit dynamic-profiles Auto-VLAN-Demux
```

2. Specify that the dynamic VLAN profile use the demux interface.

```
[edit dynamic-profiles “Auto-VLAN-Demux”]
user@host# edit interfaces demux0
```

3. Specify that the dynamic profile apply the demux interface unit value to the dynamic VLANs.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0]
user@host# edit unit $junos-interface-unit
```

4. (Optional) Specify that the demux source accepts only IPv4 (inet) packets.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-source inet
```

5. (Optional) Specify that each dynamically created interface respond to any ARP request, as long as an active route exists to the target address of the ARP request.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set proxy-arp
```

6. Specify that VLAN IDs are dynamically created.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set vlan-id $junos-vlan-id
```

7. Specify the logical underlying interface for the dynamic VLANs.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-interface-ifd-name
```

8. Specify that the VLAN demux interface can accept inet family packets for IPoE/DHCP subscribers.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

9. Specify the loopback address as the unnumbered address and preferred source address for the inet family.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit” family
inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 198.51.100.100
```

Configuring a Dynamic Profile to Dynamically Create IP Demux Interfaces

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create IP demux interfaces in the example:

```
[edit]
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit proxy-arp
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit demux-options
underlying-interface $junos-underlying-interface
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit family inet
demux-source $junos-subscriber-ip-address
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit family inet
unnumbered-address lo0.0 preferred-source-address 198.51.100.100
```

Step-by-Step Procedure

A dynamic profile that configures an IP demux interface must specify variables for unit, underlying interface name, and IP address. A dynamic IP demux interface associates specific subscribers to separate individual circuits by IP address.

To configure a dynamic profile and attach it to an interface so that it automatically creates IP demux interfaces:

1. Create a dynamic profile for dynamically creating IP demux interfaces.

```
[edit]
user@host# edit dynamic-profiles DHCP-IP-Demux
```

2. Specify that the dynamic profile use the demux0 interface.

```
[edit dynamic-profiles DHCP-IP-Demux]
user@host# edit interfaces demux0
```

3. Specify that the dynamic profile apply the interface unit value to the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0]
user@host# edit unit $junos-interface-unit
```

4. (Optional) Configure the router to respond to any ARP request, as long as the router has an active route to the target address of the ARP request.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set proxy-arp
```

5. Specify the logical underlying interface for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-underlying-interface
```

6. Specify the protocol family information for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

7. Specify the demux source address is obtained from the incoming subscriber IP address.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit” family
inet]
user@host# set demux-source $junos-subscriber-ip-address
```

8. Specify the loopback interface as the unnumbered address and the demux interface IP address as the preferred source address for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit” family
inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 198.51.100.100
```

Verification

IN THIS SECTION

Subscriber Verification

Purpose

View subscriber information on the router.

Action

- To display dynamic subscriber information:

```
user@host# show subscribers detail
```

Interface Verification

Purpose

View interface-specific information on the router.

Action

- To display interface-specific output:

```
user@host# show interfaces interface-name
```

RELATED DOCUMENTATION

Configuring Predefined Dynamic Variables in Dynamic Profiles

Dynamic 802.1Q VLAN Overview | **5**

Demultiplexing Interface Overview

CHAPTER 10

Configuring DHCP Subscriber Interfaces over Aggregated Ethernet

IN THIS CHAPTER

Static and Dynamic VLAN Subscriber Interfaces over Aggregated Ethernet Overview

IN THIS SECTION

You can configure a subscriber interface represented by a static virtual LAN (VLAN) stacked on a two-link aggregated Ethernet *logical interface*. You must configure the aggregated Ethernet logical interface on Enhanced Queuing Dense Port Concentrators (EQ DPCs) or MPC/MIC interfaces in MX Series 5G Universal Routing Platforms.

A static or dynamic VLAN subscriber interface over aggregated Ethernet can also support one-to-one active/backup link redundancy, depending on how you configure the underlying aggregated Ethernet interface.

To configure a static or dynamic VLAN subscriber interface over aggregated Ethernet, make sure you understand the following concepts.

Guidelines for Configuring an Aggregated Ethernet Logical Interface to Support a Static or Dynamic VLAN Subscriber Interface

The following guidelines for configuring an aggregated Ethernet logical interface also apply to configuring a static or dynamic VLAN subscriber interface stacked on a two-link aggregated Ethernet logical interface:

- If you need to support one-to-one active/backup link redundancy, configure the aggregated Ethernet interface in link protection mode, which requires that the two underlying physical interfaces be designated as primary and backup links.

- In addition, if you need to support one-to-one active/backup link redundancy at the DPC or MPC level, configure the aggregated Ethernet interface on physical interfaces that reside on different EQ DPCs or MPCs.

> **NOTE**: One-to-one active/backup DPC redundancy is also supported with firewall filters and policy filters for static non-VLAN interfaces configured on an aggregated Ethernet logical interfaces, provided LACP is not active.

RELATED DOCUMENTATION

Static Subscriber Interfaces and VLAN Overview | 7

Configuring a Static or Dynamic VLAN Subscriber Interface over Aggregated Ethernet | 119

Example: Configuring a Static Subscriber Interface on a VLAN Interface over Aggregated Ethernet | 123

Guidelines for Configuring Dynamic CoS for Subscriber Access

CoS for Subscriber Access Overview

Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview

IN THIS SECTION

You can configure a subscriber interface using a static or dynamic demux interface stacked on an aggregated Ethernet *logical interface*. Subscriber interfaces on static or dynamic demux interfaces can be used to identify specific subscribers (authenticated users) in an access network or to separate individual circuits. A subscriber interface on a static or dynamic demux interface over aggregated Ethernet can support one-to-one active/backup link redundancy or traffic load balancing, depending on how you configure the underlying aggregated Ethernet interface.

To configure a static or dynamic demux subscriber interface over aggregated Ethernet, make sure you understand the following concepts:

Options for Aggregated Ethernet Logical Interfaces That Support Demux Subscriber Interfaces

Traffic forwarding through a demux logical interface is dependent on the configuration of the underlying interface. Using an aggregated Ethernet interface as the underlying interface for a static or dynamic demux subscriber interface provides you with the following options:

- **1:1 Active/Backup Link Redundancy**—If you need to support one-to-one active/backup link redundancy, configure the aggregated Ethernet interface in link protection mode, which requires that two underlying physical interfaces be designated as primary and backup links. In addition, if you need to support one-to-one active/backup link redundancy at the line card level, configure the aggregated Ethernet interface on physical interfaces that reside either on different EQ DPCs or on different MPCs. When using LACP link protection, you can configure only two member links to an aggregated Ethernet interface: one active and one standby.

- **Load Balancing**—You can configure load balancing instead of 1:1 active/backup link redundancy. The Junos OS implementation of the IEEE 802.3ad standard balances traffic across the member links within an aggregated Ethernet bundle based on the Layer 3 information carried in the packet.

By default, the system supports hash-based distribution in load balancing scenarios. In this model, traffic for a logical interface can be distributed over multiple links in the aggregated Ethernet interface. If distribution flows are not even, egress CoS scheduling can be inaccurate. In addition, scheduler resources are required on every link of the aggregated Ethernet interface.

Targeted distribution enables you to target the egress traffic for IP and VLAN demux subscribers on a single member link, using a single scheduler resource. The system distributes the subscriber interfaces equally among the member links.

Hardware Requirements with Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet

IP demux subscriber interfaces over aggregated Ethernet interfaces are supported on EQ DPCs.

VLAN demux subscriber interfaces over aggregated Ethernet interfaces are supported on MX Series routers that only have MPCs installed. If the router has other line cards in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

Features Supported with Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet

Table 6 on page 117 lists key subscriber access features supported with static or dynamic demux subscriber interfaces, organized by type of underlying interface:

- Aggregated Ethernet
- Non-aggregated Ethernet (Gigabit Ethernet, Fast Ethernet, or 10-Gigabit Ethernet)

There are no feature limitations specific to demultiplexing. Instead, demux interfaces over aggregated Ethernet are subject to the same scaling and configuration limitations inherent to aggregated Ethernet logical interfaces.

Table 6: Features Supported with Static or Dynamic Demux Subscriber Interfaces

Feature	Static or Dynamic Demux Subscriber Interface	
	Aggregated Ethernet Underlying Interface	Non-aggregated Underlying Logical Interface
Protocol family support	IPv4, IPv6, and PPPoE	IPv4, IPv6, and PPPoE

Table 6: Features Supported with Static or Dynamic Demux Subscriber Interfaces *(Continued)*

Feature	Static or Dynamic Demux Subscriber Interface	
	Aggregated Ethernet Underlying Interface	Non-aggregated Underlying Logical Interface
Per-subscriber firewall filtering and statistics	Supported	Supported
Hierarchical CoS	Supported	Supported
Per-subscriber CoS parameters within the `[edit dynamic-profiles` *`profile-name`* `class-of-service]` hierarchy	Supported	Supported
Per-subscriber IGMP configuration within the `[edit dynamic-profiles` *`profile-name`* `protocols]` hierarchy **NOTE**: IP demux interfaces must use OIF mapping. See Example: Configuring Multicast with Subscriber VLANs for additional information.	Yes	Yes

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

Distribution of Demux Subscribers in an Aggregated Ethernet Interface

Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet | 120

Configuring the PPPoE Family for an Underlying Interface | 201

Example: Configuring a Static Subscriber Interface on an IP Demux Interface over Aggregated Ethernet | 127

Aggregated Ethernet Interfaces Overview

Configuring a Static or Dynamic VLAN Subscriber Interface over Aggregated Ethernet

You can configure a subscriber link represented by a static virtual LAN (VLAN) stacked on an aggregated Ethernet logical interface.

You can configure subscriber management services such as firewall filters and CoS for this subscriber interface.

To configure a subscriber interface using a static VLAN interface over an aggregated Ethernet logical interface:

1. Configure the aggregated Ethernet interface.
   a. Configure the number of aggregated Ethernet interfaces on the router.

      See Configuring the Number of Aggregated Ethernet Interfaces on the Device.

   b. Configure the aggregated Ethernet interface.

      See Configuring an Aggregated Ethernet Interface.

   c. (Optional) Configure LACP.

      See Configuring LACP for Aggregated Ethernet Interfaces.

   d. (Optional) Configure the minimum number of links.

      See Configuring Aggregated Ethernet Minimum Links.

   e. (Optional) Configure the link speed.

      See Configuring Aggregated Ethernet Link Speed.

   f. (Optional) Configure the aggregated Ethernet logical interface to support one-to-one active/backup link redundancy or traffic load balancing.

      See Configuring Aggregated Ethernet Link Protection.

      NOTE: Link protection is required if you want to configure hierarchical CoS on the aggregated Ethernet interface. For more information, see Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links.

2. Configure the static or dynamic VLAN interface.
3. Configure subscriber management services on the subscriber interface.
   - For firewall filters, see *Dynamically Attaching Statically Created Filters for Any Interface Type* or *Dynamically Attaching Statically Created Filters for a Specific Interface Family Type*.

- For hierarchical CoS, see Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links.

RELATED DOCUMENTATION

Static and Dynamic VLAN Subscriber Interfaces over Aggregated Ethernet Overview | 114

Example: Configuring a Static Subscriber Interface on a VLAN Interface over Aggregated Ethernet | 123

Guidelines for Configuring Dynamic CoS for Subscriber Access

CoS for Subscriber Access Overview

Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet

You can configure a subscriber interface using a static or dynamic IP demultiplexing (demux) logical interface stacked on an aggregated Ethernet logical interface. Optionally, you can configure the aggregated Ethernet logical interface to support one-to-one active/backup link redundancy or traffic load balancing.

1. Configure the aggregated Ethernet interface.

   a. Configure the number of aggregated Ethernet interfaces on the router.

      See Configuring the Number of Aggregated Ethernet Interfaces on the Device.

   b. Configure the aggregated Ethernet interface.

      See Configuring an Aggregated Ethernet Interface.

   c. (Optional) Configure LACP.

      See Configuring LACP for Aggregated Ethernet Interfaces.

   d. (Optional) Configure the minimum number of links.

      See Configuring Aggregated Ethernet Minimum Links.

   e. (Optional) Configure the link speed.

      See Configuring Aggregated Ethernet Link Speed.

   f. (Optional) Configure the aggregated Ethernet logical interface to support one-to-one active/backup link redundancy or traffic load balancing.

      For general instructions, see Configuring Aggregated Ethernet Link Protection.

> **NOTE**: Link protection is required if you want to configure hierarchical CoS on the aggregated Ethernet interface. For more information, see Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links.

2. Configure the aggregated Ethernet logical interface as the underlying interface to support the static or dynamic IP demux subscriber interface.

   The aggregated Ethernet interface needs to support demultiplexing of incoming traffic to the Ethernet links based on IPv4 destination or source addresses in the incoming packets. In addition, you must configure the IP address of each link.

   See Configuring an IP Demultiplexing Interface.

3. Configure the static or dynamic IP demux interface.

   > **NOTE**: IP demux interfaces currently support only the Internet Protocol version 4 (IPv4) suite (`family inet`).
   >
   > VLAN demux interfaces support the Internet Protocol version 4 (IPv4) suite (`family inet`) and the Internet Protocol version 6 (IPv6) suite (`family inet6`).
   >
   > VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has legacy DPC card in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

4. (Optional) Configure subscriber management services on the subscriber interface.

   - For firewall filters, see *Dynamically Attaching Statically Created Filters for Any Interface Type* or *Dynamically Attaching Statically Created Filters for a Specific Interface Family Type*.

   - For hierarchical CoS, see Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links.

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview | 116

Example: Configuring a Static Subscriber Interface on an IP Demux Interface over Aggregated Ethernet | 127

Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces

Configuring a Static or Dynamic VLAN Demux Subscriber Interface over Aggregated Ethernet

You can configure a subscriber interface using a static or dynamic VLAN demultiplexing (demux) logical interface stacked on an aggregated Ethernet physical interface.

1. Configure the aggregated Ethernet interface.
   a. Configure the number of aggregated Ethernet interfaces on the router.

      See Configuring the Number of Aggregated Ethernet Interfaces on the Device.

   b. Configure the aggregated Ethernet interface.

      See Configuring an Aggregated Ethernet Interface.

   c. (Optional) Configure LACP.

      See Configuring LACP for Aggregated Ethernet Interfaces.

   d. (Optional) Configure the minimum number of links.

      See Configuring Aggregated Ethernet Minimum Links.

   e. (Optional) Configure the link speed.

      See Configuring Aggregated Ethernet Link Speed.

   f. (Optional) Configure the aggregated Ethernet logical interface to support one-to-one active/backup link redundancy or traffic load balancing.

      For general instructions, see Configuring Aggregated Ethernet Link Protection.
2. Configure the aggregated Ethernet physical interface as the underlying interface to support the static or dynamic VLAN demux subscriber interface.

   The aggregated Ethernet interface needs to support demultiplexing of incoming traffic to the Ethernet links based on the VLAN ID in the incoming packets.

   See Configuring a VLAN Demultiplexing Interface.
3. Configure the static or dynamic VLAN demux interface.

   NOTE: VLAN demux interfaces support the Internet Protocol version 4 (IPv4) suite (`family inet`) and the Internet Protocol version 6 (IPv6) suite (`family inet6`).

   VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has DPC cards in addition to MPC cards, the CLI accepts the configuration, but errors are reported when the subscriber interfaces are brought up.
4. (Optional) Configure subscriber management services on the subscriber interface.

- For firewall filters, see *Dynamically Attaching Statically Created Filters for Any Interface Type* or *Dynamically Attaching Statically Created Filters for a Specific Interface Family Type*.
- For hierarchical CoS, see Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links.

RELATED DOCUMENTATION

Example: Configuring a Static Subscriber Interface on a VLAN Interface over Aggregated Ethernet

This example shows how you can configure a subscriber interface using a static virtual LAN (VLAN) stacked on a two-link aggregated Ethernet logical interface. In this example, the underlying aggregated Ethernet logical interface is configured for one-to-one active/backup redundancy at the DPC level, and per-subscriber static hierarchical class-of-service (CoS) is configured by applying CoS parameters at the aggregated Ethernet logical interface.

1. Define the number of aggregated Ethernet interfaces on the router.

   In this example, only one aggregated Ethernet logical interface is configured on the router.

```
[edit]
chassis {
    aggregated-devices {
        ethernet {
            device-count 1;
        }
    }
}
```

2. Configure `ae0`, a two-link aggregated Ethernet logical interface to serve as the underlying interface for the static VLAN subscriber interface. In order to support hierarchical CoS, the physical ports must be on EQ DPCs in MX Series routers.

   In this example, the LAG bundle is configured for one-to-one active/backup link redundancy. To support link redundancy at the DPC level, the LAG bundle attaches ports from two different EQ DPCs.

```
[edit]
interfaces {
    ge-5/0/3 {
        gigether-options {
            802.3ad {
                ae0;
                primary;
            }
        }
        ge-5/1/2 {
            gigether-options {
                802.3ad {
                    ae0;
                    backup;
                }
            }
        }
    }
}
```

3. Configure `ae0` to serve as the underlying interface for the static VLAN interface.

```
[edit]
interfaces {
    ae0 {
        hierarchical-scheduler;
        aggregated-ether-options {
            link-protection;
            minimum-links 1;
            link-speed 1g;
            lacp {
                active;
            }
```

```
        }
    }
}
```

4. Configure static traffic-shaping and scheduling parameters.

```
[edit]
class-of-service {
    forwarding-classes { # Associate queue numbers with class names
        queue 0 be;
        queue 1 e;
        queue 2 af;
        queue 3 nc;
    }
    schedulers { # Define output queue properties
        scheduler_be {
            transmit-rate percent 30;
            buffer-size percent 30;
        }
        scheduler_ef {
            transmit-rate percent 40;
            buffer-size percent 40;
        }
        scheduler_af {
            transmit-rate percent 25;
            buffer-size percent 25;
        }
        scheduler_nc {
            transmit-rate percent 5;
            buffer-size percent 5;
        }
    }
    scheduler-maps { # Associate queues with schedulers
        smap_2 {
            forwarding-class be scheduler_be;
            forwarding-class ef scheduler_ef;
            forwarding-class-af scheduler_af;
            forwarding-class-nc scheduler_nc;
        }
    }
}
```

5. Attach static CoS to the physical and logical interfaces of the aggregated Ethernet interface.

   In this example, three traffic control profiles are defined, but only two profiles are applied to the static VLAN subscriber interface over aggregated Ethernet:

   - The `tcp_for_ae_device_pir_500m` profile defines a shaping rate, and it is applied to both of the underlying physical interfaces (`ge-5/0/3` and `ge-5/1/2`).

   - The `tcp-for-ae_smap_video_pir_20m_delay_30m` profile defines a scheduler map, a shaping rate, and a delay buffer rate, and it is applied to one of the logical interfaces on the aggregated Ethernet bundle (`ae0.0`).

```
[edit]
class-of-service {
    traffic-control-profiles { # Configure traffic shaping and scheduling profiles
        tcp_for_ae_device_pir_500m {
            shaping-rate 20m;
        }
        tcp_for_ae_smap_video_pir_20m_delay_30m {
            scheduler-map smap_video;
            shaping-rate 20m;
            delay-buffer-rate 30m;
        }
        tcp_for_ae_smap_video_cir_50m_delay_75m {
            scheduler-map smap_video;
            guaranteed-rate 50m;
            delay-buffer-rate 75m;
        }
    }
    interfaces { # Apply two traffic-control profiles to the LAG
        ae0 { # Two underlying physical interfaces on separate EQ DPCs
            output-traffic-control-profile tcp-for-ae_device_pir_500m;
            unit 0 { # One of the two logical interfaces on 'ae0'
                output-traffic-control-profile tcp-for-ae_smap_video_pir_20m_delay_30m;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Example: Configuring a Static Subscriber Interface on an IP Demux Interface over Aggregated Ethernet

This example shows how you can configure a subscriber interface using a static IP demultiplexing (demux) interface stacked on a two-link aggregated Ethernet logical interface. In this example, the underlying aggregated Ethernet logical interface is configured for one-to-one active/backup redundancy at the DPC level.

1. Define the number of aggregated Ethernet interfaces on the router.

   In this example, only one aggregated Ethernet logical interface is configured on the router:

```
[edit]
chassis {
    aggregated-devices {
        ethernet {
            device-count 1;
        }
    }
}
```

2. Configure `ae0`, a two-link aggregated Ethernet logical interface to serve as the underlying interface for the static IP demux subscriber interface.

   In this example, the LAG bundle is configured for one-to-one active/backup link redundancy. To support link redundancy at the DPC level, the LAG bundle attaches ports from two different EQ DPCs.

```
[edit]
interfaces {
    ge-5/0/3 {
        gigether-options {
```

```
            802.3ad {
                ae0;
                primary;
            }
        }
    }
    ge-5/1/2 {
        gigether-options {
            802.3ad {
                ae0;
                backup;
            }
        }
    }
}
```

3. Configure the aggregated Ethernet logical interface with link protection enabled, and specify the logical demultiplexing source family type for both the active and backup links.

```
[edit]
interfaces {
    ae0 {
        aggregated-ether-options {
            link-protection;
            minimum-links 1;
            link-speed 1g;
        }
        unit 0 {
            demux-source inet {
            family inet {
                address 203.0.113.110/24;
            }
        }
        unit 1 {
            demux-source inet {
            family inet {
                address  203.0.113.111/24;
            }
        }
    }
}
```

4. Configure the IP demux interface over the aggregated Ethernet logical interface.

```
[edit]
interfaces {
    demux0 {
        unit 101 {
            demux-options {
                underlying-interface ae0.0;
            }
            family inet {
                demux-source 203.0.113.100/16;
                address 203.0.113.0/24;
            }
        }
        unit 101 {
            demux-options {
                underlying-interface ae0.1;
            }
            family inet {
                demux-source 203.0.113.221/16;
                address 203.0.113.0/24;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93
Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview | 116
Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet | 120

Example: Configuring IPv4 Static VLAN Demux Interfaces over an Aggregated Ethernet Underlying Interface with DHCP Local Server

This example shows how to configure a static IPv4 VLAN demux interface with aggregated Ethernet as the underlying interface. DHCP Local Server configuration enables the association of subscribers to the

VLAN demux interface by listing the aggregated Ethernet interface in the DHCP local server configuration.

To configure dynamic subscribers on VLAN demux interfaces:

1. Enable hierarchical scheduling and VLAN tagging on the underlying interface that you plan to use for any VLAN demux interfaces.

```
interfaces {
    ae1 {
        hierarchical-scheduler;
        vlan-tagging;
        aggregated-ether-options {
            minimum-links 1;
            lacp {
                active;
                periodic slow;
                link-protection {
                    non-revertive;
                }
            }
        }
    }
}
```

2. Define the gigabit Ethernet interfaces that are part of the aggregated Ethernet interface.

```
interfaces {
    ge-5/0/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
    ge-5/2/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
}
```

3. Define the demux interface.

```
interfaces {
    demux0 {
        unit 102 {
            proxy-arp;
            vlan-id 103;
            demux-options {
                underlying-interface ae1;
            }
            family inet {
                unnumbered-address lo0.0 preferred-source-address 173.16.1.1;
            }
        }
    }
}
```

4. Define the loopback interface.

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 127.16.1.1/32;
            }
        }
    }
}
```

5. Configure a dynamic profile for initial subscriber access.

```
dynamic-profiles {
    user-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
```

```
        protocols {
            igmp {
                interface "$junos-interface-name" {
                    version 3;
                    immediate-leave;
                    promiscuous-mode;
                }
            }
        }
    }
}
```

6. Configure the access method used to dynamically create the subscriber interfaces.

   The following stanza specifies the aggregated Ethernet interface (`ae1.0`) for use with the dynamically created subscriber interfaces.

```
system {
    services {
        dhcp-local-server {
            group myDhcpGroup {
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface ae1.0;
            }
        }
    }
}
```

   Instead of using the aggregated Ethernet interface, you can alternatively specify the specific demux interface (`demux0.102`) as the device to use with the subscriber interfaces as follows:

```
system {
    services {
        dhcp-local-server {
            group myDhcpGroup {
```

```
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface demux0.102;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148

Example: Configuring IPv4 Dynamic VLAN Demux Interfaces over an Aggregated Ethernet Underlying Interface with DHCP Local Server

This example shows how to configure the dynamic creation of IPv4 VLAN demux interfaces with aggregated Ethernet as the underlying interface. DHCP Local Server configuration enables the association of subscribers to the VLAN demux interface by listing the aggregated Ethernet interface in the DHCP local server configuration.

NOTE: VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has other cards in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

To configure dynamic subscribers on dynamic VLAN demux interfaces:

1. Enable VLAN tagging and VLAN auto-configuration on the underlying aggregated Ethernet interface that you plan to use for dynamically created VLAN demux interfaces.

```
interfaces {
    ae1 {
        vlan-tagging;
```

```
        auto-configure {
            vlan-ranges {
                dynamic-profile auto-vlanDemux-profile {
                    accept inet;
                    ranges {
                        any;
                    }
                }
            }
        }
        aggregated-ether-options {
            minimum-links 1;
            lacp {
                active;
                periodic slow;
                link-protection {
                    non-revertive;
                }
            }
        }
    }
}
```

2. Define the gigabit Ethernet interfaces that are part of the aggregated Ethernet interface.

```
interfaces {
    ge-5/0/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
    ge-5/2/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
}
```

3. Define the loopback interface.

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 127.16.1.1/32;
            }
        }
    }
}
```

4. Configure a dynamic profile for subscriber access.

```
dynamic-profiles {
    user-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
    }
}
```

5. Configure a dynamic profile for VLAN demux interface creation.

```
dynamic-profiles {
    auto-vlanDemux-profile {
        interfaces {
            demux0 {
                unit "$junos-interface-unit" {
                    vlan-id "$junos-vlan-id";
                    demux-options {
                        underlying-interface "$junos-interface-ifd-name";
                    }
                    family inet {
                        filter {
                            input rate_limit;
```

```
                            output rate_limit;
                        }
                        unnumbered-address lo0.0 preferred-source-address 127.16.1.1;
                    }
                }
            }
        }
    }
}
```

6. Configure the access method used to dynamically create the subscriber interfaces. The following stanza specifies the aggregated Ethernet interface (`ae1.0`) for use with the dynamically created subscriber interfaces.

```
system {
    services {
        dhcp-local-server {
            group myDhcpGroup {
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface ae1.0;
            }
        }
    }
}
```

Instead of using the aggregated Ethernet interface, you can alternatively specify `demux0` as the device to use with the subscriber interfaces as follows:

NOTE: Because the demux interfaces and unit values are created dynamically, the unit number is not specified for the demux0 interface.

```
system {
    services {
        dhcp-local-server {
```

```
            group myDhcpGroup {
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface demux0;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles | 101

Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148

Example: Configuring IPv6 Dynamic VLAN Demux Interfaces over an Aggregated Ethernet Underlying Interface with DHCP Local Server

This example shows how to configure the dynamic creation of IPv6 VLAN demux interfaces with aggregated Ethernet as the underlying interface. DHCP Local Server configuration enables the association of subscribers to the VLAN demux interface by listing the aggregated Ethernet interface in the DHCP local server configuration.

NOTE: VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has other cards in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

To configure dynamic subscribers on dynamic VLAN demux interfaces:

1. Enable VLAN tagging and VLAN auto-configuration on the underlying aggregated Ethernet interface that you plan to use for dynamically created VLAN demux interfaces.

```
interfaces {
    ae1 {
        vlan-tagging;
        auto-configure {
            vlan-ranges {
                dynamic-profile auto-vlanDemux-profile {
                    accept inet6;
                    ranges {
                        any;
                    }
                }
            }
        }
        aggregated-ether-options {
            minimum-links 1;
            lacp {
                active;
                periodic slow;
                link-protection {
                    non-revertive;
                }
            }
        }
    }
}
```

2. Define the gigabit Ethernet interfaces that are part of the aggregated Ethernet interface.

```
interfaces {
    ge-5/0/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
    ge-5/2/0 {
        gigether-options {
            802.3ad ae1;
        }
```

```
    }
}
```

3. Define the loopback interface.

```
interfaces {
    lo0 {
        unit 0 {
            family inet6 {
                address 2001:db8:174:1:1::1/128;
            }
        }
    }
}
```

4. Configure a dynamic profile for subscriber access.

```
dynamic-profiles {
    user-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet6;
                }
            }
        }
    }
}
```

5. Configure a dynamic profile for VLAN demux interface creation.

```
dynamic-profiles {
    auto-vlanDemux-profile {
        interfaces {
            demux0 {
                unit "$junos-interface-unit" {
                    vlan-id "$junos-vlan-id";
                    demux-options {
                        underlying-interface "$junos-interface-ifd-name";
                    }
```

```
                    family inet6 {
                        filter {
                            input v6_rate_limit;
                            output v6_rate_limit;
                        }
                        unnumbered-address lo0.0 preferred-source-address 2001:db8:174:1:1::1;
                    }
                }
            }
        }
    }
}
```

6. Configure the access method used to dynamically create the subscriber interfaces. The following stanza specifies the aggregated Ethernet interface (`ae1.0`) for use with the dynamically created subscriber interfaces.

```
system {
    services {
        dhcp-local-server {
            dhcpv6 {
                group myV6DhcpGroup {
                    authentication {
                        password test;
                        username-include {
                            user-prefix igmp-user1;
                        }
                    }
                    dynamic-profile user-profile;
                    interface ae1.0;
                }
            }
        }
    }
}
```

Instead of using the aggregated Ethernet interface, you can alternatively specify `demux0` as the device to use with the subscriber interfaces as follows:

NOTE: Because the demux interfaces and unit values are created dynamically, the unit number is not specified for the demux0 interface.

```
system {
    services {
        dhcp-local-server {
            dhcpv6 {
                group myV6DhcpGroup {
                    authentication {
                        password test;
                        username-include {
                            user-prefix igmp-user1;
                        }
                    }
                    dynamic-profile user-profile;
                    interface demux0;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles | 101
Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148

Example: Configuring IPv4 Dynamic Stacked VLAN Demux Interfaces over an Aggregated Ethernet Underlying Interface with DHCP Local Server

This example shows how to configure the dynamic creation of IPv4 stacked VLAN demux interfaces with aggregated Ethernet as the underlying interface. DHCP Local Server configuration enables the association of subscribers to the VLAN demux interface by listing the aggregated Ethernet interface in the DHCP local server configuration.

NOTE: VLAN demux subscriber interfaces over aggregated Ethernet physical interfaces are supported only for MX Series routers that have only MPCs installed. If the router has other cards in addition to MPCs, the CLI accepts the configuration but errors are reported when the subscriber interfaces are brought up.

To configure dynamic subscribers on dynamic VLAN demux interfaces:

1. Enable VLAN tagging and VLAN auto-configuration on the underlying aggregated Ethernet interface that you plan to use for dynamically created VLAN demux interfaces.

```
interfaces {
    ae1 {
        flexible-vlan-tagging;
        auto-configure {
            stacked-vlan-ranges {
                dynamic-profile auto-vlanDemux-profile {
                    accept inet;
                    ranges {
                        any;
                    }
                }
            }
        }
        aggregated-ether-options {
            minimum-links 1;
            lacp {
                active;
                periodic slow;
                link-protection {
                    non-revertive;
                }
            }
        }
    }
}
```

2. Define the gigabit Ethernet interfaces that are part of the aggregated Ethernet interface.

```
interfaces {
    ge-5/0/0 {
```

```
        gigether-options {
            802.3ad ae1;
        }
    }
    ge-5/2/0 {
        gigether-options {
            802.3ad ae1;
        }
    }
}
```

3. Define the loopback interface.

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 127.16.1.1/32;
            }
        }
    }
}
```

4. Configure a dynamic profile for subscriber access.

```
dynamic-profiles {
    user-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
    }
}
```

5. Configure a dynamic profile for VLAN demux interface creation.

```
dynamic-profiles {
    auto-vlanDemux-profile {
        interfaces {
            demux0 {
                unit "$junos-interface-unit" {
                    vlan-tags outer "$junos-stacked-vlan-id" inner "$junos-vlan-id";
                    demux-options {
                        underlying-interface "$junos-interface-ifd-name";
                    }
                    family inet {
                        filter {
                            input rate_limit;
                            output rate_limit;
                        }
                        unnumbered-address lo0.0 preferred-source-address 127.16.1.1;
                    }
                }
            }
        }
    }
}
```

6. Configure the access method used to dynamically create the subscriber interfaces. The following stanza specifies the aggregated Ethernet interface (`ae1.0`) for use with the dynamically created subscriber interfaces.

```
system {
    services {
        dhcp-local-server {
            group myDhcpGroup {
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface ae1.0;
            }
```

```
        }
    }
}
```

Instead of using the aggregated Ethernet interface, you can alternatively specify `demux0` as the device to use with the subscriber interfaces as follows:

> **NOTE**: Because the demux interfaces and unit values are created dynamically, the unit number is not specified for the demux0 interface.

```
system {
    services {
        dhcp-local-server {
            group myDhcpGroup {
                authentication {
                    password test;
                    username-include {
                        user-prefix igmp-user1;
                    }
                }
                dynamic-profile user-profile;
                interface demux0;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles | 101

Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148

CHAPTER 11

Using Dynamic Profiles to Apply Services to DHCP Subscriber Interfaces

IN THIS CHAPTER

Dynamic Profile Attachment to DHCP Subscriber Interfaces Overview

IN THIS SECTION

The router's DHCP support enables you to attach a dynamic profile to a DHCP subscriber interface. When a DHCP subscriber logs in, the router instantiates the specified dynamic profile and then applies the services defined in the profile to the interface.

You can attach dynamic profiles to all interfaces or you can specify a particular group of interfaces to which the profile is attached. Both the DHCP local server and the DHCP relay agent support the attachment of dynamic profiles to interfaces.

You can enable the following optional features when the dynamic profile is attached. The two options cannot be used together.

- Enable multiple DHCP subscribers to share the same VLAN *logical interface*. The firewall filters, CoS schedulers, and IGMP configuration of the clients are merged.
- Specify the primary dynamic profile that is instantiated when the first subscriber logs in.

Multiple DHCP Subscribers Sharing the Same VLAN Logical Interface

The `aggregate-clients` statement specifies that the router merge the firewall filters, CoS schedulers, and IGMP configuration of multiple DHCP clients that are on the same VLAN logical interface (for example, multiple clients belonging to the same household). You can configure the aggregate-clients support for all interfaces or for a group of interfaces. The `aggregate-clients` statement provides the option of either merging (chaining) or replacing software components for each client.

By default, the feature is disabled and a single DHCP client is allowed per VLAN when a dynamic profile is associated with the VLAN logical interface.

When you specify the `merge` option, the router aggregates the software components for multiple subscribers as follows:

- Firewall filters—The filters are chained together using the precedence as the order of execution. If the same *firewall filter* is attached multiple times, the filter is executed only once.
- CoS schedulers—The different CoS schedulers are merged as if the scheduler map has multiple schedulers. The merge operation for the individual traffic-control-profiles parameters (shaping-rate, delay-buffer-rate, guaranteed-rate) preserves the maximum value for each parameter.
- IGMP configuration—The current IGMP configuration is replaced with the configuration of the newest DHCP client.

When you specify the `replace` option, the entire logical interface is replaced whenever a new client logs in to the network using the same VLAN logical interface. For example, if a customer subscribes to voice, video, and data services on the network, when a voice client logs in, instead of applying a specific voice filter for only that service, the entire voice, video, and data filter chain is applied.

NOTE: You cannot use a dynamic demux interface to represent multiple subscribers in a dynamic profile attached to an interface. One dynamic demux interface represents one subscriber. Do not configure the `aggregate-clients` option when attaching a dynamic profile to a demux interface for DHCP.

Primary Dynamic Profile

The `use-primary` option enables you to specify the primary dynamic profile that is instantiated when the first subscriber logs in. Subsequent subscribers are not assigned the primary dynamic profile; instead, they are assigned the dynamic profile specified for the interface. When the first subscriber logs out, the next subscriber that logs in is assigned the primary dynamic profile.

This feature can conserve logical interfaces in a network where dynamic IP demux interfaces are used to represent subscribers. To conserve interfaces, make sure the primary profile that you specify does not create a demux interface, but provides the initial policies for the primary interface subscriber.

RELATED DOCUMENTATION

Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces | 148

Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces

IN THIS SECTION

This topic describes how to attach a dynamic profile to a DHCP subscriber interface or a DHCP client interface. When a DHCP subscriber or DHCP client logs in, the specified dynamic profile is instantiated and the services defined in the profile are applied to the interface.

This topic contains the following sections:

Attaching a Dynamic Profile to All DHCP Subscriber or All DHCP Client Interfaces

To attach a dynamic profile to all DHCP subscriber or all DHCP client interfaces:

1. At the DHCP configuration hierarchy, use the `dynamic-profile` statement to specify the name of the dynamic profile to attach to all interfaces.

   - For DHCP local server:

   ```
   [edit system services dhcp-local-server]
   user@host# set dynamic-profile vod-profile-22
   ```

   - For DHCP relay agent:

   ```
   [edit forwarding-options dhcp-relay]
   user@host# set dynamic-profile vod-profile-west
   ```

2. (Routers only) Optionally, you can configure the attribute to use when attaching the specified profile.

You can include either the `aggregate-clients` option to enable multiple DHCP subscribers to share the same VLAN logical interface, or the `use-primary` option to specify that the primary dynamic profile is used. The `aggregate-clients` option does not apply to demux subscriber interfaces. The two options are mutually exclusive.

- To enable multiple subscribers to share the same VLAN logical interface:

```
[edit system services dhcp-local-server dynamic-profile]
user@host# set aggregate-clients merge
```

- To use the primary dynamic profile:

```
[edit forwarding-options dhcp-relay dynamic-profile]
user@host# set use-primary subscriber_profile
```

Attaching a Dynamic Profile to a Group of DHCP Subscriber Interfaces or a Group of DHCP Client Interfaces

Before you begin:

- Configure the interface group.

  See *Grouping Interfaces with Common DHCP Configurations*.

To attach a dynamic profile to a group of interfaces:

1. At the DHCP configuration hierarchy, specify the name of the interface group and the dynamic profile to attach to the group.

   - For DHCP local server:

   ```
   [edit system services dhcp-local-server]
   user@host# set group boston dynamic-profile vod-profile-42
   ```

   - For DHCP relay agent:

   ```
   [edit forwarding-options dhcp-relay]
   user@host# set group quebec dynamic-profile vod-profile-east
   ```

2. (Routers only) Optionally, you can configure the attribute to use when attaching the specified profile.

You can include either the `aggregate-clients` option to enable multiple DHCP subscribers to share the same VLAN logical interface, or the `use-primary` option to specify that the primary dynamic profile is used. The `aggregate-clients` option does not apply to demux subscriber interfaces. The two options are mutually exclusive.

- To enable multiple subscribers to share the same VLAN logical interface:

```
[edit system services dhcp-local-server dynamic-profile]
user@host# set aggregate-clients merge
```

- To use the primary dynamic profile:

```
[edit forwarding-options dhcp-relay dynamic-profile]
user@host# set use-primary subscriber_profile
```

RELATED DOCUMENTATION

Dynamic Profiles Overview

Dynamic Profile Attachment to DHCP Subscriber Interfaces Overview | **146**

CHAPTER 12

Configuring DHCP IP Demux and PPPoE Demux Interfaces Over the Same VLAN

IN THIS CHAPTER

Example: Concurrent Configuration of Dynamic DHCP IP Demux and PPPoE Demux Interfaces over the Same VLAN Demux Interface

IN THIS SECTION

This example shows how to configure both dynamic DHCP IP demux and PPPoE demux interfaces over the same dynamic VLAN demux interface. The example provides an IPv4 configuration. However, you can also configure concurrent IP over Ethernet/DHCP and PPPoE interfaces over the same VLAN interface using IPv6 addressing.

Requirements

Before you begin, make sure to configure either DHCP Relay or DHCP Local Server. For information about configuring either of these components, see *Extended DHCP Relay Agent Overview* or *Understanding Differences Between Legacy DHCP and Extended DHCP*.

Overview

With the introduction of the `family pppoe` statement, PPPoE is no longer treated as an exclusive encapsulation configuration and you can configure VLAN interfaces with multiple protocol interface stacks. For example, you can configure IP over Ethernet/DHCP and PPPoE interfaces concurrently over a single VLAN interface.

Configuration

IN THIS SECTION

Preparing a Subscriber Access Interface

CLI Quick Configuration

To quickly configure the aggregated Ethernet interface over which subscribers access the router:

```
[edit]
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-5/0/9 gigether-options 802.3ad ae0
set interfaces ge-5/1/9 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux accept any
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux accept any line-
identity includes circuit-id
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux accept any line-
identity includes remote-id
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux accept any line-
identity includes accept-no-ids mac-address
set interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux ranges ranges
1000-1500
```

```
set interfaces ae0 auto-configure stacked-vlan-ranges dynamic-profile Auto-Stacked-VLAN-Demux
accept any
set interfaces ae0 auto-configure stacked-vlan-ranges dynamic-profile Auto-Stacked-VLAN-Demux
ranges 1501-2000,any
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp link-protection
```

Step-by-Step Procedure

When configuring multiple protocol interface stacks concurrently over the same VLAN interface, you must configure physical interfaces over which DHCP or PPPoE clients initially access the router. We recommend that you specify the same VLAN tagging for the interface that you expect from incoming clients. This example uses flexible VLAN tagging to simultaneously support transmission of 802.1Q VLAN single-tag and dual-tag frames on logical interfaces on the same Ethernet port.

To automatically create dynamic VLANs, the interface must also include the VLAN range type (single or stacked), dynamic profile reference, and any specific ranges you want the VLANs to use.

To configure a physical interface for subscriber access:

1. Access the physical interface over which you want subscribers to initially access the router.

   ```
   [edit]
   user@host# edit interfaces ge-5/0/9
   ```

2. Specify the aggregated Ethernet interface to which the physical interface belongs.

   ```
   [edit interfaces ge-5/0/9]
   user@host# set gigether-options 802.3ad ae0
   ```

3. Repeat Step 1 and Step 2for each interface you want to assign to the aggregated Ethernet bundle.

   ```
   [edit]
   user@host# set interfaces ge-5/1/9 gigether-options 802.3ad ae0
   ```

4. Access the aggregated Ethernet interface.

   ```
   [edit]
   user@host# edit interfaces ae0
   ```

5. Specify the VLAN tagging that you want the aggregated Ethernet interfaces to use.

```
[edit interfaces ae0]
user@host# set flexible-vlan-tagging
```

6. Edit the `auto-configure` stanza to automatically configure VLANs.

```
[edit interfaces ae0]
user@host# edit auto-configure
```

7. Edit the `vlan-ranges` stanza for single-tagged VLANs.

```
[edit interfaces ae0 auto-configure]
user@host# edit vlan-ranges
```

8. Specify the dynamic VLAN profile that you want the interface to use for dynamically creating single-tagged VLANs.

```
[edit interfaces ae0 auto-configure vlan-ranges]
user@host# edit dynamic-profile Auto-VLAN-Demux
```

9. Specify what VLAN Ethernet packet type the VLAN profile accepts.

```
[edit interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux]
user@host# set accept any
```

10. Specify the VLAN ranges that you want the dynamic profile to use. The following example specifies a lower VLAN ID limit of 1000 and an upper VLAN ID limit of 1500.

```
[edit interfaces ae0 auto-configure vlan-ranges dynamic-profile Auto-VLAN-Demux]
user@host# set ranges 1000-1500
```

11. Edit the `stacked-vlan-ranges` stanza for the dual-tagged VLANs.

```
[edit interfaces ae0 auto-configure]
user@host# edit stacked-vlan-ranges
```

12. Specify the dynamic VLAN profile that you want the interface to use for dynamically creating dual-tagged VLANs.

```
[edit interfaces ae0 auto-configure stacked-vlan-ranges]
user@host# edit dynamic-profile Auto-Stacked-VLAN-Demux
```

13. Specify what VLAN Ethernet packet type the stacked VLAN profile accepts.

```
[edit interfaces ae0 auto-configure stacked-vlan-ranges dynamic-profile Auto-Stacked-VLAN-Demux]
user@host# set accept any
```

14. Specify the outer and inner stacked VLAN ranges that you want the dynamic profile to use. The following example specifies an outer stacked VLAN ID range from 1501 through 2000 (to avoid overlapping VLAN IDs with single-tag VLANs) and an inner stacked VLAN ID range of any (enabling a range from 1 through 4094 for the inner stacked VLAN ID).

```
[edit interfaces ge-5/0/9 auto-configure stacked-vlan-ranges dynamic-profile Auto-Stacked-VLAN-Demux]
user@host# set ranges 1501-2000,any
```

15. (Optional) Activate the transmission of LACP packets on the aggregated Ethernet interfaces.

```
[edit interfaces ae0]
user@host# set aggregated-ether-options lacp active
```

16. Specify that the aggregated Ethernet interfaces use link protection.

```
[edit interfaces ae0]
user@host# set aggregated-ether-options link-protection
```

Preparing the Loopback Interface

CLI Quick Configuration

To quickly configure the required loopback interface for this example:

```
[edit]
set interfaces lo0.0 unit 0 family inet address 100.100.100.1/32
```

Step-by-Step Procedure

You must configure a loopback interface for use as the unnumbered address and preferred source address for dynamically created interfaces.

To configure the required loopback interface for this example:

1. Configure a loopback interface.

   ```
   [edit]
   user@host# edit interfaces lo0.0
   ```

2. Specify that the loopback interface accept inet packets.

   ```
   [edit interfaces lo0 unit 0]
   user@host# edit family inet
   ```

3. Specify the IP address for the loopback interface.

   ```
   [edit interfaces lo0 unit 0 family inet]
   user@host# set address 100.100.100.1/32
   ```

Configuring a Dynamic Profile to Create Dynamic Single-Tagged VLANs

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create single-tagged VLANs in the example:

```
[edit]
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux-source
inet
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit proxy-arp
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-id $junos-
vlan-id
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-id $junos-
vlan-id auto-configure line-identity includes circuit-id
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-id $junos-
vlan-id auto-configure line-identity includes remote-id
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-id $junos-
vlan-id auto-configure line-identity includes accept-no-ids mac-address
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux options
underlying-interface $junos-interface-ifd-name
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit family inet
unnumbered-address lo0.0 preferred source-address 100.100.100.1
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit family pppoe
duplicate-protection
set dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit $junos-interface-unit family pppoe
dynamic-profile PPP-Base-PAP
```

Step-by-Step Procedure

For both dynamic DHCP IP demux and dynamic PPPoE interfaces to reside concurrently on a single-tagged VLAN interface, the VLAN interface must first exist.

To configure a dynamic profile that automatically creates VLAN interfaces:

1. Create a dynamic profile for automatically creating VLAN interfaces.

   ```
   [edit]
   user@host# edit dynamic-profiles Auto-VLAN-Demux
   ```

2. Specify that the dynamic VLAN profile use the demux interface.

```
[edit dynamic-profiles “Auto-VLAN-Demux”]
user@host# edit interfaces demux0
```

3. Specify that the dynamic profile apply the demux interface unit value to the dynamic VLANs.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0]
user@host# edit unit $junos-interface-unit
```

4. Specify that the demux source accept IPv4 (inet) packets.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-source inet
```

5. (Optional) Specify that each dynamically created interface respond to any ARP request, as long as an active route exists to the target address of the ARP request.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set proxy-arp
```

6. Specify that VLAN IDs are dynamically created.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set vlan-id $junos-vlan-id
```

7. (Optional)

   To automatically configure the dynamic profile to accept PPPoE packets with line identifier as circuit ID, remote ID, or use mac-address if no identifiers are received using the `accept-no-ids mac-address` option.

   If the `accept-no-ids` attribute is configured and if the PPPoE PADI is received without ACI or ARI string, the source mac-address is concatenated to the underlying-interface and the concatenated string is used in the creation of ACI VLANs.

```
set auto-configure line-identity includes circuit-id
```

```
set auto-configure line-identity includes remote-id
set auto-configure line-identity includes accept-no-ids mac-address
```

8. Specify the logical underlying interface for the dynamic VLANs.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-interface-ifd-name
```

9. Specify that the VLAN demux interface can accept inet family packets for IP over Ethernet/DHCP subscribers.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

10. Specify the loopback address as the unnumbered address and preferred source address for the inet family.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”
family inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 100.100.100.1
```

11. Specify that the VLAN demux interface can accept pppoe family packets for PPPoE subscribers.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family pppoe
```

12. Prevent multiple PPPoE sessions from being created for the same PPPoE subscriber on the same VLAN interface.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”
family pppoe]
user@host# set duplicate-protection
```

13. Apply the dynamic PPP interface profile to any dynamic PPP interfaces.

```
[edit dynamic-profiles Auto-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”
family pppoe]
user@host# set dynamic-profile PPP-Base-PAP
```

14. (Optional)

    To automatically configure the dynamic PPP interface profile to accept PPPoE packets with line identifier as circuit ID, remote ID, or use mac-address if no identifiers are received using the `accept-no-ids mac-address` option.

    If the `accept-no-ids` attribute is configured and if the PPPoE PADI is received without ACI or ARI string, the source mac-address is concatenated to the underlying-interface and the concatenated string is used in the creation of ACI VLANs.

```
set accept pppoe line-identity includes circuit-id
set accept pppoe line-identity includes remote-id
set accept pppoe line-identity includes accept-no-ids mac-address
```

Configuring a Dynamic Profile to Create Dynamic Dual-Tagged VLANs

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create stacked/dual-tagged VLANs in the example:

```
[edit]
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux-
source inet
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit proxy-
arp
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-
tags outer $junos-stacked-vlan-id
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-
tags inner $junos-vlan-id
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-
tags inner $junos-vlan-id accept any line-identity includes circuit-id
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-
```

```
tags inner $junos-vlan-id accept any line-identity includes remote-id
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit vlan-
tags inner $junos-vlan-id accept any line-identity includes accept-no-ids mac-address
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit demux
options underlying-interface $junos-interface-ifd-name
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit family
inet unnumbered-address lo0.0 preferred source-address 100.100.100.1
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit family
pppoe duplicate-protection
set dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit $junos-interface-unit family
pppoe dynamic-profile PPP-Base-PAP
```

Step-by-Step Procedure

For both dynamic DHCP IP demux and dynamic PPPoE interfaces to reside concurrently on a VLAN interface, the VLAN interface must first exist.

To configure a dynamic profile that automatically creates stacked/dual-tagged VLAN interfaces:

1. Create a dynamic profile for automatically creating VLAN interfaces.

   ```
   [edit]
   user@host# edit dynamic-profiles Auto-Stacked-VLAN-Demux
   ```

2. Specify that the dynamic VLAN profile use the demux interface.

   ```
   [edit dynamic-profiles “Auto-Stacked-VLAN-Demux”]
   user@host# edit interfaces demux0
   ```

3. Specify that the dynamic profile apply the demux interface unit value to the dynamic VLANs.

   ```
   [edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0]
   user@host# edit unit $junos-interface-unit
   ```

4. Specify that the demux source accept IPv4 (inet) packets.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-
unit”]
user@host# set demux-source inet
```

5. (Optional) Specify that each dynamically created interface respond to any ARP request, as long as an active route exists to the target address of the ARP request.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-
unit”]
user@host# set proxy-arp
```

6. Specify that the outer VLAN ID is dynamically created.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-
unit”]
user@host# set vlan-id -tags outer $junos-stacked-vlan-id
```

7. Specify that the inner VLAN ID is dynamically created.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-
unit”]
user@host# set vlan-id -tags inner $junos-vlan-id
```

8. (Optional)

   To automatically configure the dynamic profile to accept PPPoE packets with line identifier as circuit ID, remote ID, or use mac-address if no identifiers are received using the `accept-no-ids mac-address` option.

   If the `accept-no-ids` attribute is configured and if the PPPoE PADI is received without ACI or ARI string, the source mac-address is concatenated to the underlying-interface and the concatenated string is used in the creation of ACI VLANs.

```
set auto-configure line-identity includes circuit-id
set auto-configure line-identity includes remote-id
```

```
set auto-configure line-identity includes accept-no-ids mac-address
```

9. Specify the logical underlying interface for the dynamic VLANs.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-interface-ifd-name
```

10. Specify that the VLAN demux interface can accept inet family packets for IP over Ethernet/DHCP subscribers.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

11. Specify the loopback address as the unnumbered address and preferred source address for the inet family.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-unit” family inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 100.100.100.1
```

12. Specify that the VLAN demux interface can accept pppoe family packets for PPPoE subscribers.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family pppoe
```

13. Prevent the activation of another dynamic PPPoE logical interface on the same demux underlying interface.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit “$junos-interface-unit” family pppoe]
user@host# set duplicate-protection
```

14. Apply the dynamic PPP interface profile to any dynamic PPP interfaces.

```
[edit dynamic-profiles Auto-Stacked-VLAN-Demux interfaces demux0 unit "$junos-interface-
unit" family pppoe]
user@host# set dynamic-profile PPP-Base-PAP
```

15. (Optional)

    To automatically configure the dynamic PPP interface profile to accept PPPoE packets with line identifier as circuit ID, remote ID, or use mac-address if no identifiers are received using the `accept-no-ids mac-address` option.

    If the `accept-no-ids` attribute is configured and if the PPPoE PADI is received without ACI or ARI string, the source mac-address is concatenated to the underlying-interface and the concatenated string is used in the creation of ACI VLANs.

```
set accept pppoe line-identity includes circuit-id
set accept pppoe line-identity includes remote-id
set accept pppoe line-identity includes accept-no-ids mac-address
```

Configuring a Dynamic Profile to Create Dynamic IP Demux Interfaces

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create DHCP IP demux interfaces in the example:

```
[edit]
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit proxy-arp
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit demux-options
underlying-interface $junos-underlying-interface
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit family inet
demux-source $junos-subscriber-ip-address
set dynamic-profiles DHCP-IP-Demux interfaces demux0 unit $junos-interface-unit family inet
unnumbered-address lo0.0 preferred-source-address 100.100.100.1
```

Step-by-Step Procedure

To configure a dynamic profile that automatically creates IP demux interfaces:

1. Create a dynamic profile for dynamically creating IP demux interfaces.

```
[edit]
user@host# edit dynamic-profiles DHCP-IP-Demux
```

2. Specify that the dynamic profile use the demux0 interface.

```
[edit dynamic-profiles DHCP-IP-Demux]
user@host# edit interfaces demux0
```

3. Specify that the dynamic profile apply the interface unit value to the dynamic PPPoE interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0]
user@host# edit unit $junos-interface-unit
```

4. (Optional) Configure the router to respond to any ARP request, as long as the router has an active route to the target address of the ARP request.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set proxy-arp
```

5. Specify the logical underlying interface for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# set demux-options underlying-interface $junos-underlying-interface
```

6. Specify the protocol family information for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

7. Specify the demux source address is obtained from the incoming subscriber IP address.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit "$junos-interface-unit" family
inet]
user@host# set demux-source $junos-subscriber-ip-address
```

8. Specify the loopback interface as the unnumbered address and the demux interface IP address as the preferred source address for the dynamic IP demux interfaces.

```
[edit dynamic-profiles DHCP-IP-Demux interfaces demux0 unit "$junos-interface-unit" family
inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 100.100.100.1
```

Configuring a Dynamic Profile to Create Dynamic PPPoE Interfaces

CLI Quick Configuration

To quickly configure the dynamic profile used to dynamically create PPPoE interfaces in the example:

```
[edit]
set dynamic-profiles PPP-Base-PAP interfaces pp0 unit $junos-interface-unit ppp-options pap
set dynamic-profiles PPP-Base-PAP interfaces pp0 unit $junos-interface-unit pppoe-options
underlying-interface $junos-underlying-interface server
set dynamic-profiles PPP-Base-PAP interfaces pp0 unit $junos-interface-unit no-keepalives
set dynamic-profiles PPP-Base-PAP interfaces pp0 unit $junos-interface-unit family inet
unnumbered-address lo0.0
```

Step-by-Step Procedure

1. Create a dynamic profile for automatically creating PPPoE interfaces.

```
[edit]
user@host# edit dynamic-profiles PPP-Base-PAP
```

2. Specify that the dynamic PPPoE profile use the pp0 interface.

```
[edit dynamic-profiles PPP-Base-PAP]
user@host# edit interfaces pp0
```

3. Specify that the dynamic profile apply the interface unit value to the dynamic PPPoE interfaces.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0]
user@host# edit unit $junos-interface-unit
```

4. Specify that dynamically created PPPoE interfaces use PAP authentication.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# set ppp-options pap
```

5. Specify the logical underlying interface for the dynamic PPPoE interfaces.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# set pppoe-options underlying-interface $junos-underlying-interface
```

6. Specify that the router act as a PPPoE server.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# set pppoe-options server
```

7. (Optional) Disable the sending of keepalive messages on the dynamic PPPoE interfaces.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# set no-keepalives
```

8. Specify the protocol family information for the dynamic PPPoE interfaces.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# edit family inet
```

9. Specify the loopback interface as the unnumbered address for the dynamic PPPoE interfaces.

```
[edit dynamic-profiles PPP-Base-PAP interfaces pp0 unit “$junos-interface-unit”]
user@host# set unnumbered-address lo0.0
```

Verification

IN THIS SECTION

Subscriber Verification

Purpose

View subscriber information on the router.

Action

- To display dynamic subscriber information:

```
user@host# show subscribers detail
```

Interface Verification

Purpose

View interface-specific information on the router.

Action

- To display interface-specific output:

```
user@host# show interfaces interface-name
```

RELATED DOCUMENTATION

Configuring a Basic Dynamic Profile

Configuring Predefined Dynamic Variables in Dynamic Profiles

Dynamic 802.1Q VLAN Overview | **5**

Demultiplexing Interface Overview

Configuring the PPPoE Family for an Underlying Interface | **201**

CHAPTER 13

Providing Security for DHCP Interfaces Using MAC Address Validation

IN THIS CHAPTER

MAC Address Validation for Subscriber Interfaces Overview

IN THIS SECTION

MAC address validation enables the router to validate that received packets contain a trusted IP source and an Ethernet MAC source address.

Configuring MAC address validation can provide additional validation when subscribers access billable services. MAC address validation provides additional security by enabling the router to drop packets that do not match, such as packets with spoofed addresses.

When subscribers log in, they are automatically assigned IP addresses by DHCP. With MAC address validation enabled, the router compares the IP source and MAC source addresses against trusted addresses, and forwards or drops the packets according to the match and the validation mode.

You can enable MAC address validation on interfaces configured with IPv4 or IPv6 addresses.

Supported Types of Subscriber Interfaces

MAC address validation is supported on statically or dynamically created Ethernet interfaces and demux interfaces as follows:

- When the router is configured for a normal (non-enhanced) network services mode, MAC address validation is supported on both DPCs and MPCs. The router can be populated completely with one or the other type of line card, or have a mix of both types. Normal network services mode is the default.
- When the router is configured for Enhanced IP Network Services mode or Enhanced Ethernet Network Services mode, MAC address validation is supported only on MPCs. If the router has both DPCs and MPCs, or only DPCs, you cannot configure the chassis to be in enhanced mode.

MAC address validation is optimized for scaling when the router is in enhanced network services modes. Enhanced network services modes affect other features, such as multicast and firewall filters, so you must take that in to consideration when deciding whether to configure enhanced mode. For more information about the enhanced network service modes, see Network Services Mode Overview.

In normal network services mode, you can use the `show interfaces statistics` *`interface-name`* command to display a per-interface count of the packets that failed validation and were dropped. In enhanced network services mode, this command does not count the dropped packets; you must contact Juniper Networks Customer Support for assistance in collecting this data.

Trusted Addresses

A trusted address tuple is a 32–bit/128-bit IP address and a 48–bit MAC address. Prefixes and ranges are not supported.

The IP source address and the MAC source address used for validation must be from a trusted source.

All static ARP addresses configured through the CLI are trusted addresses; dynamic ARP addresses are not considered trusted addresses.

Addresses dynamically created through an extended DHCP local server or extended DHCP relay are also trusted addresses. When a DHCP server and client negotiate an IP address, the resulting IP address and MAC address tuple is trusted. Each DHCP subscriber can generate more than one address tuple.

Each MAC address can have more than one IP address, which can result in more than one valid tuple. Each IP address must map to one MAC address.

Types of MAC Address Validation

You can configure either of two types or modes of MAC address validation, loose or strict. The behavior of the two modes varies depending on how well the incoming packets match the trusted address tuples.

The modes differ only when the IP source address alone does not match any trusted IP address. Table 7 on page 172 compares the behavior of the two modes. Dropped packets are considered to be spoofed.

Table 7: Comparison of MAC Address Validation Modes

Incoming Packet Addresses Match Trusted Address Tuple	Loose Mode Action	Strict Mode Action
• IP source address matches and • MAC source address matches	Forwards packet	Forwards packet
• IP source address matches but • MAC source address does not match	Drops packet	Drops packet
• IP source address does not match and • MAC source address either matches or does not match	Forwards packet	Drops packet

Configuring strict mode is a more conservative strategy because it requires both received source addresses to match trusted addresses.

When you configure MAC address validation for IP demux interfaces in a dynamic profile and specify either loose or strict validation, the resulting behavior is always loose validation. To enable strict behavior for a dynamic IP demux interface, you must configure strict validation for both the IP demux interface and the underlying interface.

RELATED DOCUMENTATION

Configuring MAC Address Validation for Subscriber Interfaces | 173

Configuring MAC Address Validation for Subscriber Interfaces

IN THIS SECTION

This topic describes how to configure MAC address validation for subscriber interfaces in dynamic profiles.

The subscriber interfaces can be statically created and associated with a dynamic profile (for example, VLAN interfaces) or dynamically created in the dynamic profile (such as demux interfaces).

By default, MAC address validation is disabled.

This topic contains the following sections:

Configuring MAC Address Validation for Static Subscriber Interfaces

This topic describes how to configure MAC address validation for static subscriber interfaces.

The following configuration example uses IPv4 address and family `inet` in the configuration. For interfaces with IPv6 address, use family `inet6` in the configuration.

Before you begin:

- Configure the dynamic profile.

  See *Configuring a Basic Dynamic Profile*.

- (Optional) Configure an enhanced network services mode.

  See Configuring Junos OS to Run a Specific Network Services Mode in MX Series Routers.

To configure MAC address validation on static subscriber interfaces:

1. Configure the static VLAN interface.

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family inet
```

2. Configure the type of MAC address validation for the interface.

- To configure loose validation:

```
[edit interfaces interface-name unit logical-unit-number family inet]
user@host# set mac-validate loose
```

- To configure strict validation:

```
[edit interfaces interface-name unit logical-unit-number family inet]
user@host# set mac-validate strict
```

For example, to configure loose validation on interface fe-0/0/0.0, configure the following:

```
[edit interfaces fe-0/0/0 unit 0 family inet]
user@host# set mac-validate loose
```

After you configure MAC address validation, associate the static VLAN interface with the dynamic profile.

Configuring MAC Address Validation for Dynamic Subscriber Interfaces

This topic describes how to configure MAC address validation for subscriber interfaces created on demux interfaces in dynamic profiles.

When you configure MAC address validation for demux interfaces in a dynamic profile and specify either `loose` or `strict` validation, the resulting behavior is always loose validation. To enable strict behavior for a dynamic IP demux interface, besides configuring either `loose` or `strict` mode on the IP demux interface, you must also configure strict validation on the underlying interface.

The following configuration examples use IPv4 address and family `inet` in the configuration. For interfaces with IPv6 address, use family `inet6` in the configuration.

Before you begin:

- Configure the dynamic profile.

  See *Configuring a Basic Dynamic Profile*.

- Configure the dynamic IP demux interface.
- (Optional) Configure an enhanced network services mode.

  See Configuring Junos OS to Run a Specific Network Services Mode in MX Series Routers.

To configure loose MAC address validation for a dynamic subscriber interface:

- Configure loose validation for the demux interface.

```
[edit dynamic-profiles profile-name interfaces demux0 unit “$junos-interface-unit” family
inet]
user@host# set mac-validate loose
```

  For loose validation, you do not need to configure MAC address validation on the underlying interface.

To configure strict MAC address validation for a dynamic subscriber interface:

1. Configure validation for the demux interface.

```
[edit dynamic-profiles profile-name interfaces demux0 unit “$junos-interface-unit” family
inet]
user@host# set mac-validate validation-mode
```

   NOTE: Remember, although you must configure validation on the IP demux interface, it does not matter which mode you specify because the behavior is always loose.

2. Configure strict validation for the underlying interface.

```
[edit interfaces interface-name unit logical-unit-number family inet]
user@host# set mac-validate strict
```

   The underlying interface in this case is statically configured—for example, ge-1/0/0.1—and assigned to a DHCP configuration group that is associated with the dynamic profile. In a more complicated configuration, the underlying interface itself can be configured by a dynamic profile; in that case the validation is configured in the profile that creates the underlying interface.

SEE ALSO

Subscriber Interfaces and Demultiplexing Overview | 93

RELATED DOCUMENTATION

MAC Address Validation for Subscriber Interfaces Overview | 170

CHAPTER 14

RADIUS-Sourced Weights for Targeted Distribution

IN THIS CHAPTER

RADIUS-Sourced Weights for Interface and Interface Set Targeted Distribution

IN THIS SECTION

Targeted distribution is a way to load balance traffic between the member links of an aggregated Ethernet bundle by distributing the logical interfaces or interface sets across the links. Egress traffic for a subscriber is targeted for a single member link, making it possible to use a single CoS scheduler for the subscriber to optimize resource use.

Interfaces and interface sets are assigned to primary and backup member links to yield an even distribution of subscribers across all member links.

- A link is selected as primary when it is up and has the lightest subscriber load. If no links are up then the available link with the lightest subscriber load is selected.
- A link is selected as backup when it is the available link with the lightest subscriber load. The redundancy mode configured for the aggregated Ethernet bundle affects the pool of available links. For example, module redundancy excludes all links on the same module from being assigned as backup.

The subscriber load is also known as the link weight. You can configure an explicit weight for targeted subscribers based on factors important to you, such as CoS or bandwidth requirement. The member links are assigned based on the value of the weight. The weight is configured per dynamic profile for an interface or interface set. Starting in Junos OS Release 18.4R1, you can have RADIUS supply the weight value per subscriber. To do so, specify either of the following predefined variables that corresponds to the relevant RADIUS VSA conveyed in the Access-Accept message when a subscriber is authenticated.

- $junos-interface-target-weight corresponds to VSA 26-214, Interface-Targeting-Weight.
- $junos-interface-set-target-weight corresponds to VSA 26-213, Interface-Set-Targeting-Weight.

Diameter AVPs 213 and 214 can be used for the same purpose during NASREQ processing.

When you use a dynamic interface set with targeted distribution, the interface set and its member interfaces are assigned to the same aggregated Ethernet member link. This means that you have to configure targeted distribution for both the interface set and its member interfaces. The dynamic interface set is created when the first member interface is instantiated. The weight that is used to associate the interface set and its members to the aggregated Ethernet member link is either of the following:

- The weight assigned to the interface set. The interface set weight is either explicitly configured or sourced from RADIUS VSA 26-214 when the first member interface is authorized.
- The weight assigned to the first member interface. The interface weight is used when the interface set has no assigned weight. The weight for the first member interface is either explicitly configured or sourced from RADIUS VSA 26-213 when the first member interface is authorized.

BEST PRACTICE: Always ensure that a weight is assigned to the interface set by the CLI configuration or by RADIUS.

Because the weight of the first instantiated member interface can provide the weight for the interface set, the weights of subsequent member interfaces have no effect on the assignment of the interface set and its members to a given aggregated Ethernet member link.

BEST PRACTICE: We recommend that the weight assigned to the interface set be representative of the member interfaces to ensure optimal distribution among the aggregated Ethernet member links. Consequently, there is no advantage to sourcing weights from RADIUS for both the interface set and its member interfaces, because sourcing the weight for only the interface set is sufficient.

The RADIUS-sourced weight for an interface set cannot change after the set is created when the first member interface is authorized. Consequently, only interfaces having the same weight as the first interface can become members of the interface set. Consider the following example:

1. Suppose that when the first dynamic subscriber interface is authorized, the authorization from RADIUS includes VSA 26-214 with a value of 100.

2. The interface set is then assigned a weight of 100 based on the first interface weight.

3. When the second dynamic subscriber interface is authorized, the authorization includes VSA 26-214 with a value of 200.

4. Because the weight for the interface set cannot change; it remains at 100 and the instantiation of the subscriber session on the second interface fails.

Benefits of RADIUS-Sourced Weighting

- Enables per-subscriber weighting based on RADIUS user record, rather than per dynamic profile.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, you can have RADIUS supply the weight value per subscriber.

RELATED DOCUMENTATION

Using RADIUS-Sourced Weights for Interface and Interface Set Targeted Distribution | 178

Understanding Support for Targeted Distribution of Logical Interface Sets of Static VLANs over Aggregated Ethernet Logical Interfaces

Using RADIUS-Sourced Weights for Interface and Interface Set Targeted Distribution

Instead of explicitly configuring a subscriber weight for targeted distribution of interfaces and interface sets across aggregated Ethernet member links, you can use predefined variables to extract the weight value provided by RADIUS in one of two VSAs conveyed in the Access-Accept message when the subscriber is authenticated.

- $junos-interface-target-weight corresponds to VSA 26-214, Interface-Targeting-Weight.
- $junos-interface-set-target-weight corresponds to VSA 26-213, Interface-Set-Targeting-Weight.

When you use a dynamic interface set with targeted distribution, the interface set and its member interfaces are assigned to the same aggregated Ethernet member link. This means that you have to configure targeted distribution for both the interface set and its member interfaces. The dynamic interface set is created when the first member interface is instantiated.

To derive the interface target weight from RADIUS:

1. Configure your RADIUS server to provide the desired value for VSA 26-214. Consult your RADIUS server documentation for more information.
2. Configure targeted distribution for the interface.

```
[edit dynamic-profiles profile-name interfaces demux0 unit $junos-interface-unit ]
user@host# set targeted-distribution
```

3. Specify the interface target predefined variable.

```
[edit dynamic-profiles profile-name interfaces demux0 unit $junos-interface-unit]
user@host# set targeted-options weight $junos-interface-target-weight
```

4. (Optional) Configure a default value in case VSA 26-214 is not received in the Access-Accept message.

```
[edit dynamic-profiles profile-name predefined-variable-defaults]
user@host# set interface-target-weight weight-value
```

To derive the interface set target weight from RADIUS:

1. Configure your RADIUS server to provide the desired value for VSA 26-213. Consult your RADIUS server documentation for more information.

2. Configure targeted distribution for the interface set.

```
[edit dynamic-profiles profile-name interfaces interface-set $junos-svlan-interface-set-name]
user@host# set targeted-distribution
```

3. Specify the interface target predefined variable.

```
[edit dynamic-profiles profile-name interfaces interface-set $junos-svlan-interface-set-name]
user@host# set targeted-options weight $junos-interface-set-target-weight
```

4. (Optional) Configure a default value in case VSA 26-213 is not received in the Access-Accept message.

```
[edit dynamic-profiles profile-name predefined-variable-defaults]
user@host# set interface-set-target-weight weight-value
```

RELATED DOCUMENTATION

RADIUS-Sourced Weights for Interface and Interface Set Targeted Distribution | 176

CHAPTER 15

Verifying Configuration and Status of Dynamic Subscribers

IN THIS CHAPTER

Verifying Configuration and Status of Dynamic Subscribers and Associated Sessions, Services, and Firewall Filters

IN THIS SECTION

Purpose

Verify configuration and status of dynamic subscribers, sessions, services, and firewall filters.

You can display information about subscribers in different ways, depending on the options you use with the `show subscriber` command. You can use details from one set of output with another command to display more detailed information of interest.

Action

- To display basic information for all subscribers:

```
user@host> show subscribers
Interface IP Address/VLAN ID User Name LS:RI
demux0.1073741824 0x8100.1500 0x8100.2900 user@test.com default:testnet
demux0.1073741825 0x8100.1500 0x8100.2901 user@test.com default:testnet
demux0.1073741826 0x8100.1500 0x8100.2902 user@test.com default:testnet
demux0.1073741827 0x8100.1500 0x8100.2903 user@test.com default:testnet
demux0.1073741826 172.16.200.6 user@test.com default:testnet
demux0.1073741827 172.16.200.7 user@test.com default:testnet
demux0.1073741824 172.16.200.8 user@test.com default:testnet
demux0.1073741825 172.16.200.9 user@test.com default:testnet
demux0.1073741828 0x8100.1500 0x8100.2910 user@test.com default:default
demux0.1073741828 20.20.0.2 user@test.com default:default
```

- To display more detailed information about a particular subscriber interface:

```
user@host> show subscribers interface demux0.1073741826 extensive
Type: VLAN
User Name: user@test.com
Logical System: default
Routing Instance: testnet
Interface: demux0.1073741826
Interface type: Dynamic
Dynamic Profile Name: profile-vdemux-relay-23qos
MAC Address: 00:00:5e:00:53:04
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: user@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: 00:00:5e:00:53:04
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
```

- To display traffic information for firewall filters.

```
user@host> show firewall
...
Filter: content-cb-in-demux0.1073741826-in
Counters:
Name                 Bytes     Packets
__junos-dyn-service-counter     84336      1004

Filter: content-cb-out-demux0.1073741826-out
Counters:
Name                 Bytes   Packets
__junos-dyn-service-counter        0          0
...
```

Instead of issuing successive commands to track the details for one subscriber interface, you can choose to display detailed information for all subscribers. However, the more subscribers you have, the more tedious it becomes to look through all the results for particular items of interest.

- To display detailed information for all subscribers:

```
user@host> show subscribers detail
```

```
user@host> show subscribers extensive
```

Meaning

The output examples in this section show increasingly detailed information about dynamically created subscriber interfaces, including how many there are, what they are, and their characteristics; how many service sessions are active and what they are; whether firewall filters are attached to the sessions and what those filters are; and how much, if any, traffic is being filtered.

In the sample output shown here, the `show subscriber` command lists all the subscriber logical interfaces, including demux0.1073741826. You then display details about that interface and its associated subscribers with the `show subscribers interface demux0.1073741826 extensive` command. The Service Session Name fields for service sessions 25 and 26 in that output show two services are active on the interface, SUB-QOS and service-cb-content. The IPv4 Input Filter Name and the IPv4 Output Filter Name fields show that two filters have been applied to the service-cb-content session: content-cb-in-demux0.1073741826-in and content-cb-out-demux0.1073741826-out. You then use the `show firewalls` command to list the filters and see how much, if any, traffic is being filtered.

RELATED DOCUMENTATION

CLI Explorer

3

PART

Configuring PPPoE Subscriber Interfaces

CHAPTER 16

Configuring Dynamic PPPoE Subscriber Interfaces

IN THIS CHAPTER

Subscriber Interfaces and PPPoE Overview

IN THIS SECTION

You can configure the router to dynamically create Point-to-Point Protocol over Ethernet (PPPoE) logical interfaces on statically created underlying Ethernet interfaces. The router creates the dynamic interface in response to the receipt of a PPPoE Active Discovery Request (PADR) control packet on the underlying interface. Because the router creates a dynamic PPPoE *logical interface* on demand when a subscriber logs in to the network, dynamic PPPoE logical interfaces are also referred to as *dynamic PPPoE subscriber interfaces*.

This overview covers the following topics:

Benefits of Using Dynamic PPPoE Subscriber Interfaces

Configuring and using dynamic PPPoE subscriber interfaces offers the following benefits:

- On-demand dynamic interface creation

  Dynamic PPPoE subscriber interfaces provides the flexibility of dynamically creating the PPPoE subscriber interface only when needed; that is, when a subscriber logs in on the associated underlying Ethernet interface. By contrast, statically created interfaces allocate and consume system resources when the interface is created. Configuring and using dynamically created interfaces helps you effectively and conveniently manage edge or access networks in which large numbers of subscribers are constantly logging in to and logging out from the network on a transient basis.

- Dynamic removal of PPPoE subscriber interfaces without manual intervention

  When the PPPoE subscriber logs out or the PPPoE session is terminated, the router dynamically deletes the associated PPPoE subscriber interface without your intervention, thereby restoring any consumed resources to the router.

- Use of dynamic profiles to efficiently manage multiple subscriber interfaces

  By using a profile, you reduce the management of a large number of interfaces by applying a set of common characteristics to multiple interfaces. When you configure a dynamic profile for PPPoE, you use predefined dynamic variables in the profile to represent information that varies from subscriber to subscriber, such as the logical unit number and underlying interface name. These variables are dynamically replaced with the values supplied by the network when the subscriber logs in.

- Denial of service (DoS) protection

  You can configure the underlying Ethernet interface with certain PPPoE-specific attributes that can reduce the potential for DoS attacks. Duplicate protection, which is disabled by default, prevents activation of another dynamic PPPoE logical interface on the underlying interface when a PPPoE logical interface for the same client is already active on the underlying interface. You can also specify the maximum number of PPPoE sessions that the router can activate on the underlying interface. By enabling duplicate protection and restricting the maximum number of PPPoE sessions on the underlying interface, you can ensure that a single toxic PPPoE client cannot monopolize allocation of the PPPoE session.

- Support for dynamic PPPoE subscriber interface creation from PPPoE service name tables

You can assign a previously configured PPPoE dynamic profile to a named, `empty`, or `any` service entry in a PPPoE service name table, or to an agent circuit identifier/agent remote identifier (ACI/ARI) pair defined for these services. The router uses the attributes defined in the profile to instantiate a dynamic PPPoE subscriber interface based on the service name, ACI, and ARI information provided by the PPPoE client during PPPoE negotiation. To specify the routing instance in which to instantiate the dynamic PPPoE subscriber interface, you can assign a previously configured routing instance to a named, `empty`, or `any` service, or to an ACI/ARI pair defined for these services. The dynamic profile and routing instance configured for the PPPoE service name table overrides the dynamic profile and routing instance assigned to the PPPoE underlying interface on which the dynamic subscriber interface is created.

Supported Platforms for Dynamic PPPoE Subscriber Interfaces

Configuration of dynamic PPPoE subscriber interfaces over static underlying Ethernet interfaces is supported on MPC/MIC interfaces on MX Series 5G Universal Routing Platforms.

Sequence of Operations for PPPoE Subscriber Access

When a PPPoE subscriber logs in the PPPoE protocol defines the sequence of operations by which a connection is established and traffic flow is enabled on the dynamic PPPoE subscriber interface. Similarly, when the PPPoE subscriber logs out from the network, PPPoE defines the sequence that occurs to terminate the connection and remove the dynamic PPPoE subscriber interface from the router.

The router creates a dynamic PPPoE subscriber interface for each new PPPoE session, and removes the dynamic PPPoE subscriber interface when the session is terminated due to subscriber logout, PPP negotiation failure, or down status of the underlying Ethernet interface. Dynamic PPPoE subscriber interfaces are never reused for multiple PPPoE sessions.

Sequence When a PPPoE Subscriber Logs In

In a PPPoE subscriber network, the router acts as a *remote access concentrator*, also known as a *PPPoE server*. For a PPPoE client to initiate a PPPoE session with a PPPoE server, it must first perform PPPoE Discovery to identify the Ethernet MAC address of the remote access concentrator that can service its request. Based on the network topology, there may be more than one remote access concentrator with which the client can communicate. The Discovery process enables a PPPoE client to find all remote access concentrators and then select one to connect to.

The following sequence occurs when a PPPoE subscriber logs in to the network. Steps 1 through 5 in this sequence are part of the PPPoE Discovery process.

1. The PPPoE client broadcasts a PPPoE Active Discovery Initiation (PADI) packet to all remote access concentrators in the network.

2. One or more remote access concentrators respond to the PADI packet by sending a PPPoE Active Discovery Offer (PADO) packet, indicating that they can service the client request. The PADO packet includes the name of the access concentrator from which it was sent.

3. The client sends a unicast PPPoE Active Discovery Request (PADR) packet to the access concentrator it selects.

4. On receipt of the PADR packet on the underlying interface associated with a PPPoE dynamic profile, the router uses the attributes configured in the dynamic profile to create the dynamic PPPoE logical interface.

5. The router sends a PPPoE Active Discovery Session (PADS) packet to confirm establishment of the PPPoE connection.

6. The PPP Link Control Protocol (LCP) negotiates the PPP link between the client and the PPPoE server.

7. The subscriber is authenticated using the PPP authentication protocol (CHAP or PAP) configured in the PPPoE dynamic profile.

8. The PPP Network Control Protocol (NCP) negotiates the IP routing protocol and network family.

9. The PPP server issues an IP access address for the client, and the router adds the client access route to its routing table.

10. The router instantiates the dynamic profile and applies the attributes configured in the profile to the dynamic PPPoE subscriber interface.

11. PPP NCP negotiation completes, enabling traffic flow between the PPPoE client and the PPPoE server.

Sequence When a PPPoE Subscriber Logs Out

The following sequence occurs when a PPPoE subscriber logs out of the network:

1. The client terminates the PPP connection and the router receives an LCP termination request.

2. The router removes the client access router from its routing table.

3. The router sends or receives a PPPoE Active Discovery Termination (PADT) packet to end the PPPoE connection.

4. The router deactivates the subscriber, gathers final statistics for the PPPoE session, and sends the RADIUS server an Acct-Stop accounting message.

5. The router de-instantiates the PPPoE dynamic profile and removes the PPPoE logical interface. The router does not reuse the PPPoE logical interface for future dynamic PPPoE sessions.

Revival of PPPoE Sessions on Backup Routers (MX240, MX304, MX480, MX960, MX10004, MX10008, MX2010, and MX2020 Devices)

The enhanced PPPoE sessions creation functionality supports faster switchover of subscriber PPPoE sessions from an active MX router to a standby MX router.

After failover, the newly active standby router starts receiving traffic from subscribers. The standby router requires at least one unknown session ID packet or control packet from a subscriber, to re-establish the connection.

The PPPoE session can switchover without waiting for the keepalive time to expire. When the standby router receives an unknown session ID packet from a PPP session, it responds with PPPoE Active Discovery Termination (PADT) message. This PADT response causes the PPP client to restart immediately, which reduces the time to restart PPP sessions.

Additionally the following features are supported:

- PPPoE subscribers can quickly re-establish PPP sessions on routers.
- Rate limiting of the number of data and control packets with unknown session ID sent to routers.
- Smooth transition to reconnect all PPPoE subscribers during multiple failovers.

Table 8: Change History for Subscriber Interfaces and PPPoE

Release	Description
23.2 Junos OS Evolved	Starting from Junos release 23.2R1, the enhanced PPPoE sessions creation functionality supports faster switchover of subscriber PPPoE sessions from an active MX router to a standby MX router.

Table 9: Platform Support for Subscriber Interfaces and PPPoE

Platform	Difference
MX240, MX304, MX480, MX960, MX10004, MX10008, MX2010, and MX2020 Devices	**Revival of PPPoE Sessions on Backup Routers:** The enhanced PPPoE sessions creation functionality supports faster switchover of subscriber PPPoE sessions from an active MX router to a standby MX routers.

RELATED DOCUMENTATION

Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191
Configuring Dynamic PPPoE Subscriber Interfaces | 194

Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview

IN THIS SECTION

Creating a dynamic PPPoE subscriber interface over a static underlying Ethernet interface consists of two basic steps:

1. Configure a dynamic profile to define the attributes of the PPPoE *logical interface*.

2. Attach the dynamic profile to a statically created underlying Ethernet interface configured with PPPoE encapsulation.

This overview describes the concepts you need to understand to configure a dynamic PPPoE subscriber interface, and covers the following topics:

PPPoE Dynamic Profile Configuration

You use predefined dynamic variables in the PPPoE dynamic profile to represent information that varies from subscriber to subscriber, such as the logical unit number and underlying interface name. These variables are dynamically replaced with the values supplied by the network when the subscriber logs in. On receipt of traffic on an underlying Ethernet interface to which a dynamic profile is attached, the router creates the dynamic PPPoE logical interface, also referred to as a *dynamic PPPoE subscriber interface*, on the underlying interface and applies the properties configured in the dynamic profile.

To provide basic access for PPPoE subscribers, the dynamic profile must provide a minimal configuration for a `pp0` (PPPoE) logical interface that includes at least the following attributes:

- The logical unit number, represented by the `$junos-interface-unit` predefined dynamic variable

- The name of the underlying Ethernet interface, represented by the `$junos-underlying-interface` predefined dynamic variable
- Configuration of the router to act as a PPPoE server
- The PPP authentication protocol (PAP or CHAP)
- The unnumbered address for the `inet` (IPv4) or `inet6` (IPv6) protocol family

You can also optionally configure additional options for PPPoE subscriber access in the dynamic profile, including:

- The keepalive interval, or the option to disable sending keepalive messages
- The IPv4 or IPv6 address of the dynamic PPPoE logical interface
- The service sets and filters, input filters, and output filters to be applied to the dynamic PPPoE logical interface

PPPoE Underlying Interface Configuration

After you configure a dynamic profile to define the attributes of a dynamic PPPoE subscriber interface, you must attach the dynamic profile to the underlying Ethernet interface on which you want the router to dynamically create the PPPoE logical interface. The underlying interface for a dynamic PPPoE logical interface must be statically created and configured with PPPoE (`ppp-over-ether`) encapsulation. When a PPPoE subscriber logs in on the underlying interface, the router dynamically creates the PPPoE logical interface and applies the attributes defined in the profile to the interface.

In addition to attaching the dynamic profile to the interface, you can also configure the underlying interface with one or more of the following optional PPPoE-specific attributes:

- Prevention of another dynamic PPPoE logical interface from being activated on the underlying interface when a PPPoE logical interface for a client with the same MAC address is already active on that interface
- Maximum number of dynamic PPPoE logical interfaces (sessions) that the router can activate on the underlying interface
- An alternative access concentrator name in the AC-NAME tag in a PPPoE control packet

Address Assignment for Dynamic PPPoE Subscriber Interfaces

If the subscriber address for a dynamic PPPoE interface is not specified by means of the Framed-IP-Address (8) or Framed-Pool (88) RADIUS IETF attributes during authentication, the router allocates an IP address from the first IPv4 local address-assignment pool defined in the routing instance. For this reason, make sure that the local address assigned for the `inet` (IPv4) address family is in the same subnet as the addresses obtained from the first IPv4 local address-assignment pool.

The router allocates the IP address from the first IPv4 local address-assignment pool under either of the following conditions:

- RADIUS returns no address attributes.
- RADIUS authentication does not take place because only address allocation is requested.

If the first IPv4 local address-assignment pool has no available addresses, or if no IPv4 local address-assignment pools are configured, the router does not allocate an IP address to the dynamic PPPoE subscriber interface, and denies access to the associated subscriber. To avoid depletion of IP addresses, you can configure linked address-assignment pools on the first IPv4 local address-assignment pool to create one or more backup pools.

For more information, see *Address-Assignment Pool Configuration Overview*.

Guidelines for Configuring Dynamic PPPoE Subscriber Interfaces

Observe the following guidelines when you configure dynamic PPPoE subscriber interfaces:

- You can configure dynamic PPPoE subscriber interfaces for the `inet` (IPv4) and `inet6` (IPv6) protocol families.
- When you configure the `pp0` (PPPoE) logical interface in a PPPoE dynamic profile, you must include the `pppoe-options` subhierarchy at the `[edit dynamic-profiles` *`profile-name`* `interfaces pp0 unit "$junos-interface-unit"]` hierarchy level. At a minimum, the `pppoe-options` subhierarchy must include the name of the underlying Ethernet interface, represented by the `$junos-underlying-interface` predefined dynamic variable, and the `server` statement, which configures the router to act as a PPPoE server. If you omit the `pppoe-options` subhierarchy from the configuration, the `commit` operation fails.
- When you configure CHAP or PAP authentication in a PPPoE dynamic profile, you cannot configure additional options for the `chap` or `pap` statements. This is because the router supports only unidirectional authentication for dynamic interfaces; that is, the router always functions as the authenticator.
- When you attach the PPPoE dynamic profile to an underlying Ethernet interface, ensure that both of the following conditions are met:
  - The PPPoE dynamic profile has already been configured on the router.
  - The underlying Ethernet interface has already been statically configured on the router with PPPoE (`ppp-over-ether`) encapsulation.

- You cannot attach a PPPoE dynamic profile to an underlying Ethernet interface that is already associated with static PPPoE logical interfaces. Conversely, you cannot associate static PPPoE logical interfaces with an underlying Ethernet interface that already has a PPPoE dynamic profile attached.

RELATED DOCUMENTATION

Subscriber Interfaces and PPPoE Overview | 186
Configuring Dynamic PPPoE Subscriber Interfaces | 194
Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Gigabit Ethernet VLAN Interface | 203
Understanding PPPoE Service Name Tables | 260

Configuring Dynamic PPPoE Subscriber Interfaces

To enable the router to create a dynamic PPPoE subscriber interface on a PPPoE underlying interface, you define the attributes of the PPPoE logical interface in a dynamic profile, and then configure the underlying interface to use the dynamic profile.

To configure a dynamic PPPoE subscriber interface:

1. Configure a dynamic profile to define the attributes of the PPPoE logical interface.

   See "Configuring a PPPoE Dynamic Profile" on page 195.
2. Configure the underlying Ethernet interface to use the dynamic profile for PPPoE.

   See "Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces" on page 198.
3. (Optional) Assign a dynamic profile and routing instance to a service name or ACI/ARI pair in a PPPoE service name table to instantiate a dynamic PPPoE subscriber interface based on the information provided by the PPPoE client.

   See "Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation" on page 275.
4. (Optional) Verify the dynamic PPPoE configuration by displaying or clearing PPPoE session statistics, and displaying information about the underlying Ethernet interface and PPPoE logical interface.

   See "Verifying and Managing Dynamic PPPoE Configuration" on page 292.

RELATED DOCUMENTATION

Subscriber Interfaces and PPPoE Overview | 186
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Configuring a PPPoE Dynamic Profile

You can configure a basic dynamic profile for PPPoE subscribers that defines the attributes of the dynamic PPPoE logical subscriber interface (pp0).

To configure a basic PPPoE dynamic profile:

1. Name the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles basic-pppoe-profile
```

2. Specify that you want to configure the `pp0` logical interface in the dynamic profile.

```
[edit dynamic-profiles basic-pppoe-profile]
user@host# edit interfaces pp0
```

3. Specify `$junos-interface-unit` as the predefined variable to represent the logical unit number for the `pp0` interface.

   The `$junos-interface-unit` variable is replaced with the actual unit number supplied by the network when the subscriber logs in.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0]
user@host# edit unit $junos-interface-unit
```

4. Configure PPPoE-specific options for the `pp0` interface.

   a. Specify the `$junos-underlying-interface` predefined variable to represent the name of the underlying Ethernet interface on which the router creates the dynamic PPPoE logical interface.

      The `$junos-underlying-interface` variable is replaced with the actual name of the underlying interface supplied by the network when the subscriber logs in.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set pppoe-options underlying-interface $junos-underlying-interface
```

b. Configure the router to act as a PPPoE server, also known as a remote access concentrator.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set pppoe-options server
```

5. Configure the PPP authentication protocol for the `pp0` interface.

   For dynamic interfaces, the router supports only unidirectional authentication; that is, the router always functions as the authenticator. When you configure PPP authentication in a dynamic profile, the **chap** and **pap** statements do not support any additional configuration options.

   - To configure CHAP authentication:

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

   - To configure PAP authentication:

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

6. Modify the keepalive interval, or configure the router to disable sending keepalive messages.

   - To modify the keepalive interval:

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set keepalives interval 15
```

   - To disable sending keepalive messages:

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set no-keepalives
```

7. Configure the protocol family for the `pp0` interface.

   a. Specify that you want to configure the `inet` (IPv4) or `inet6` (IPv6) protocol family.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# edit family inet
```

b. Specify the IPv4 or IPv6 address of the dynamic PPPoE logical interface.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"
family inet]
user@host# set address 6.6.6.7/32
```

c. Configure the unnumbered address for the protocol family.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"
family  inet]
user@host# set unnumbered-address lo0.0
```

d. Specify the input and output service sets that you want to apply to the dynamic PPPoE logical interface.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"
family inet]
user@host# set service input service-set inputService_100
user@host# set service input post-service-filter postService_20
user@host# set service output service-set outputService_200
```

e. Specify the input and output filters that you want to apply to the dynamic PPPoE logical interface.

   To control the order in which filters are processed, you can optionally specify a precedence value for the input filter, output filter, or both.

```
[edit dynamic-profiles basic-pppoe-profile interfaces pp0 unit "$junos-interface-unit"
family inet]
user@host# set filter input pppoe-input-filter
user@host# set filter output pppoe-output-filter precedence 50
```

RELATED DOCUMENTATION

Subscriber Interfaces and PPPoE Overview | 186

Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Gigabit Ethernet VLAN Interface | 203

Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces

After you configure a dynamic profile to define the attributes of a dynamic PPPoE subscriber interface, you must attach the dynamic profile to a statically created underlying Ethernet interface.

Before you begin:

1. Configure the static underlying Ethernet interface on which you want the router to dynamically create the PPPoE logical interface.

   For information about configuring static Ethernet interfaces, see Configuring Ethernet Physical Interface Properties.

2. Configure a PPPoE dynamic profile.

   - See "Configuring a PPPoE Dynamic Profile" on page 195.

To configure an underlying Ethernet interface for a dynamic PPPoE subscriber interface:

1. Specify the name and logical unit number of the static underlying Ethernet interface to which you want to attach the PPPoE dynamic profile.

```
[edit]
user@host# edit interfaces ge-1/0/1 unit 0
```

2. Configure PPPoE encapsulation on the underlying interface.

```
[edit interfaces ge-1/0/1 unit 0]
user@host# set encapsulation ppp-over-ether
```

3. Specify that you want to configure PPPoE-specific options on the underlying interface.

```
[edit interfaces ge-1/0/1 unit 0]
user@host# edit pppoe-underlying-options
```

4. Attach a previously configured PPPoE dynamic profile to the underlying interface.

You cannot attach a PPPoE dynamic profile to an underlying Ethernet interface that is already associated with static PPPoE logical interfaces. Conversely, you cannot associate static PPPoE logical interfaces with an underlying Ethernet interface that already has a PPPoE dynamic profile attached.

```
[edit interfaces ge-1/0/1 unit 0 pppoe-underlying-options]
user@host# set dynamic-profile basic-pppoe-profile
```

5. (Optional) Enable duplicate protection to prevent another dynamic PPPoE logical interface from being activated on the underlying interface when a PPPoE logical interface for a client with the same MAC address is already active on that interface.

```
[edit interfaces ge-1/0/1 unit 0 pppoe-underlying-options]
user@host# set duplicate-protection
```

6. (Optional) Specify the alternative name for the access concentrator, also known as the PPPoE server, in the AC-NAME tag in a PPPoE control packet

```
[edit interfaces ge-1/0/1 unit 0 pppoe-underlying-options]
user@host# set access-concentrator server-east
```

RELATED DOCUMENTATION

Subscriber Interfaces and PPPoE Overview | 186
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191
Configuring Dynamic PPPoE Subscriber Interfaces | 194
Configuring the PPPoE Family for an Underlying Interface | 201
Configuring Lockout of PPPoE Subscriber Sessions | 247
Verifying and Managing Dynamic PPPoE Configuration | 292
Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Gigabit Ethernet VLAN Interface | 203
Configuring Ethernet Physical Interface Properties

bgp_over_dynamic_pppoe

SUMMARY

IN THIS SECTION

•

•

•

1.

2.

3.

Benefits of <feature-name>

•

•

•

WHAT's NEXT

Configuring the PPPoE Family for an Underlying Interface

Before you begin, configure the underlying interface. When you want to configure PPPoE on an aggregated Ethernet bundle, you must configure the PPPoE family over a VLAN demux interface as an intermediate underlying option. The VLAN demux interface can be static or dynamic.

You can configure the PPPoE family on an underlying interface as an alternative to configuring PPPoE encapsulation on that interface. You cannot configure both on the same interface. You can configure the same attributes for the PPPoE family as you can for an interface configured with `pppoe-underlying-options`.

To configure the PPPoE family over an underlying interface:

1. Specify the PPPoE family.

```
[edit interfaces demux0 unit logical-unit-number]
user@host# set family pppoe
```

2. (Optional) Configure an alternative access concentrator name to be used instead of the system name in PPPoE control packets for the dynamic PPPoE subscriber interface.

```
[edit interfaces demux0 unit logical-unit-number family pppoe]
user@host# set access-concentrator name
```

3. (Optional) Attach a dynamic profile to determine the properties of the dynamic PPPoE logical interface when it is created.

```
[edit interfaces demux0 unit logical-unit-number family pppoe]
user@host# set dynamic-profile profile-name
```

RELATED DOCUMENTATION

Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview | 116

Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198

Configuring Lockout of PPPoE Subscriber Sessions | 247

Example: Configuring a Static PPPoE Subscriber Interface on a Static Underlying VLAN Demux Interface over Aggregated Ethernet | 206

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Underlying VLAN Demux Interface over Aggregated Ethernet | 215

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Dynamic Underlying VLAN Demux Interface over Aggregated Ethernet | 223

Ignoring DSL Forum VSAs from Directly Connected Devices

When CPE devices are directly connected to a BNG, you may want the router to ignore any DSL Forum VSAs that it receives in PPPoE control packets because the VSAs can be spoofed by malicious subscribers. Spoofing is particularly serious when the targeted VSAs are used to authenticate the subscriber, such as Agent-Circuit-Id [26-1] and Agent-Remote-ID [26-2]. You can include the `direct-connect` statement to ignore DSL Forum VSAs on static or dynamic PPPoE interfaces or PPPoE underlying interfaces.

To configure the router to ignore DSL Forum VSAs on specific PPPoE interfaces:

1. Specify that you want to configure PPPoE-specific options on the interface:

   - For a PPPoE family in a dynamic profile for a VLAN demultiplexing (demux) logical interface:

     ```
     [edit dynamic-profiles profile-name interfaces demux0 unit logical-unit-number]
     user@host# edit family pppoe
     ```

   - For a PPPoE family in a dynamic profile:

     ```
     [edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
     user@host# edit family pppoe
     ```

   - For a PPPoE underlying interface in a dynamic profile:

     ```
     [edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
     user@host# edit pppoe-underlying-options
     ```

   - For a PPPoE family on an underlying interface:

     ```
     [edit interfaces interface-name unit logical-unit-number]
     user@host# edit family pppoe
     ```

- For an underlying interface with PPPoE encapsulation:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit pppoe-underlying-options
```

2. Specify that the router ignores DSL forum VSAs received on a specific interface.

```
[edit ... family pppoe]
user@host# set direct-connect
```

or

```
[edit ... pppoe-underlying-options]
user@host# set direct-connect
```

RELATED DOCUMENTATION

Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198

Configuring the PPPoE Family for an Underlying Interface | 201

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Gigabit Ethernet VLAN Interface

This example shows how to configure a dynamic PPPoE subscriber interface on a statically configured Gigabit Ethernet VLAN underlying interface. When a PPPoE subscriber logs in on the underlying interface, the router creates the dynamic PPPoE subscriber interface with the attributes specified in the dynamic profile.

In this example, the dynamic PPPoE profile, `pppoe-profile-east`, defines options for PPPoE subscribers accessing the network, and includes the predefined dynamic variables `$junos-interface-unit`, which represents the logical unit number of the dynamic PPPoE logical interface, and `$junos-underlying-interface`, which represents the name of the underlying Ethernet interface. The `pppoe-profile-east` dynamic profile is assigned to the underlying Ethernet VLAN interface `ge-2/0/3.1` that is configured with PPPoE (`ppp-over-ether`) encapsulation.

When the router dynamically creates the PPPoE subscriber interface on `ge-2/0/3.1` in response to a subscriber login, the values of `$junos-interface-unit` and `$junos-underlying-interface` are dynamically replaced with the actual logical unit number and interface name, respectively, that are supplied by the network when the PPPoE subscriber logs in.

To configure a dynamic PPPoE subscriber interface:

1. Configure a dynamic profile to define the attributes of the dynamic PPPoE subscriber interface.

```
[edit]
dynamic-profiles {
    pppoe-profile-east {
        interfaces {
            pp0 {
                unit "$junos-interface-unit" {
                    ppp-options {
                        chap;
                    }
                    pppoe-options {
                        underlying-interface "$junos-underlying-interface";
                        server;
                    }
                    keepalives interval 30;
                    family inet {
                        filter {
                            input pppoe-input-filter-east;
                            output pppoe-output-filter-east precedence 20;
                        }
                        service {
                            input {
                                service-set inputService-east;
                                post-service-filter postService-east;
                            }
                            output {
                                service-set outputService-east;
                            }
                        }
                        address 127.0.1.2/32;
                        unnumbered-address lo0.0;
                    }
                }
            }
        }
```

```
    }
}
```

2. Assign the dynamic PPPoE profile to the static underlying Ethernet interface, and define PPPoE-specific attributes for the underlying interface.

```
[edit]
interfaces {
    ge-2/0/3 {
        vlan-tagging;
        unit 1 {
            encapsulation ppp-over-ether;
            vlan-id 100;
            pppoe-underlying-options {
                access-concentrator server-east;
                duplicate-protection;
                dynamic-profile pppoe-profile-east;
                max-sessions 10;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Subscriber Interfaces and PPPoE Overview | 186
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191
Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198

CHAPTER 17

Configuring PPPoE Subscriber Interfaces over Aggregated Ethernet Examples

IN THIS CHAPTER

Example: Configuring a Static PPPoE Subscriber Interface on a Static Underlying VLAN Demux Interface over Aggregated Ethernet

IN THIS SECTION

This example shows how you can configure static PPPoE subscriber interfaces over aggregated Ethernet bundles to provide subscriber link redundancy.

Requirements

PPPoE over VLAN demux interfaces over aggregated Ethernet requires the following hardware and software:

- MX Series 5G Universal Routing Platforms
- MPCs
- Junos OS Release 11.2 or later

No special configuration beyond device initialization is required before you can configure this feature.

Overview

Aggregated Ethernet bundles enable link redundancy between the router and networking devices connected by Ethernet links. This example describes how to configure link redundancy for static PPPoE subscribers over aggregated Ethernet interface with an intermediate static VLAN demux interface. Sample tasks include configuring a two-member aggregated Ethernet bundle on `ae0`, configuring a static VLAN demux interface, `demux0.100`, that underlies the PPPoE subscriber interface, `pp0.100`, and configuring the PPPoE subscriber interface including characteristics of the PPPoE family.

This example does not show all possible configuration choices.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure link redundancy for static PPPoE subscribers over a static VLAN demux interface over aggregated Ethernet, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
[edit]
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-5/0/3 gigether-options 802.3ad ae0
set interfaces ge-5/0/3 gigether-options 802.3ad primary
set interfaces ge-5/1/2 gigether-options 802.3ad ae0
set interfaces ge-5/1/2 gigether-options 802.3ad backup
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 aggregated-ether-options link-protection
```

```
edit interfaces demux0 unit 100
set vlan-id 100
set demux-options underlying-interface ae0
set family pppoe access-concentrator pppoe-server-1
set family pppoe duplicate-protection
set family pppoe max-sessions 16000
top
edit interfaces pp0 unit 100
set pppoe-options underlying-interface demux0.100
set pppoe-options server
set family inet unnumbered-address lo0.0
top
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure link redundancy for static PPPoE subscribers over a static VLAN demux interface over aggregated Ethernet:

1. Define the number of aggregated Ethernet devices on the router.

   ```
   [edit chassis]
   user@host# set aggregated-devices ethernet device-count 1
   ```

2. Configure a two-link aggregated Ethernet logical interface to serve as the underlying interface for the static VLAN demux subscriber interface. In this example, the LAG bundle is configured for one-to-one active/backup link redundancy. To support link redundancy at the MPC level, the LAG bundle attaches to ports from two different MPCs.

   ```
   [edit interfaces]
   user@host# set ge-5/0/3 gigether-options 802.3ad ae0
   user@host# set ge-5/0/3 gigether-options 802.3ad primary
   user@host# set ge-5/1/2 gigether-options 802.3ad ae0
   user@host# set ge-5/1/2 gigether-options 802.3ad backup
   ```

3. Enable link protection on the aggregated Ethernet logical interface and configure support for single and dual (stacked) VLAN tags.

```
[edit interfaces]
user@host# set ae0 aggregated-ether-options link-protection
user@host# set ae0 flexible-vlan-tagging
```

4. Configure the VLAN demux interface over the aggregated Ethernet logical interface.

```
[edit interfaces]
user@host# set demux0 unit 100 vlan-id 100
user@host# set demux0 unit 100 demux-options underlying-interface ae0
```

5. Configure the PPPoE family attributes on the VLAN demux interface.

```
[edit interfaces]
user@host# set demxu0 unit 100 family pppoe access-concentrator pppoe-server-1
user@host# set demux0 unit 100 family pppoe duplicate-protection
user@host# set demux0 unit 100 family pppoe max-sessions 16000
```

6. Configure the VLAN demux interface as the underlying interface on which the PPPoE logical interface is created.

```
[edit interfaces]
user@host# set pp0 unit 100 pppoe-options underlying-interface demux0.100
user@host# set pp0 unit 100 pppoe-options server
user@host# set pp0 unit 100 family inet unnumbered-address lo0.0
```

Results

From configuration mode, confirm the aggregated device configuration by entering the `show chassis` command. Confirm the interface configuration by entering the `show interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis
```

```
aggregated-devices {
    ethernet {
        device-count 1;
    }
}
```

```
[edit]
user@host# show interfaces
    ge-5/0/3 {
        gigether-options {
            802.3ad {
                ae0;
                primary;
            }
        }
    }
    ge-5/1/2 {
        gigether-options {
            802.3ad {
                ae0;
                backup;
            }
        }
    }
    ae0 {
        flexible-vlan-tagging;
        aggregated-ether-options {
            link-protection;
        }
    }
    demux0 {
        unit 100 {
            vlan-id 100;
            demux-options {
                underlying-interface ae0;
            }
            family pppoe {
                access-concentrator pppoe-server-1;
                duplicate-protection;
                max-sessions 16000;
            }
```

```
        }
    }
    pp0 {
        unit 100 {
            pppoe-options {
                underlying-interface demux0.100;
                server;
            }
            family inet {
                unnumbered-address lo0.0;
            }
        }
    }
```

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

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the Aggregated Ethernet Interface Configuration

Purpose

Verify that the interface values match your configuration, the link is up, and traffic is flowing.

Action

From operational mode, enter the `show interfaces redundancy` command.

```
user@host> show interfaces redundancy
```

```
Interface  State           Last change  Primary      Secondary    Current status
ae0        On primary                   ge-5/0/3     ge-5/1/2     both up
```

From operational mode, enter the `show interfaces ae0` command.

```
user@host> show interfaces ae0
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 606
  Link-level type: Ethernet, MTU: 1522, Speed: 1Gbps, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 1, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:00:5e:00:53:d0, Hardware address: 00:00:5e:00:53:d0
  Last flapped   : 2011-03-11 13:24:18 PST (2d 03:34 ago)
  Input rate     : 1984 bps (2 pps)
  Output rate    : 0 bps (0 pps)

  Logical interface ae0.32767 (Index 69) (SNMP ifIndex 709)
    Flags: SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :        371259          2      46036116         1984
        Output:             0          0             0            0
    Protocol multiservice, MTU: Unlimited
      Flags: Is-Primary
```

Meaning

The `show interfaces redundancy` output shows the redundant link configuration and that both link interfaces are up. The `show interfaces ae0` output shows that the aggregated Ethernet interface is up and that traffic is being received on the logical interface.

Verifying the demux0 Interface Configuration

Purpose

Verify that the VLAN demux interface displays the configured PPPoE family attributes and the member links in the aggregated Ethernet bundle.

Action

From operational mode, enter the `show interfaces demux0` command.

```
user@host> show interfaces demux0.100
Logical interface demux0.100 (Index 76) (SNMP ifIndex 61160)
    Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 ]
    Encapsulation: ENET2
    Demux:
      Underlying interface: ae0 (Index 199)
    Link:
      ge-5/0/3
      ge-5/1/2
    Input packets : 2
    Output packets: 18575
    Protocol pppoe
      Dynamic Profile: none,
      Service Name Table: None,
      Max Sessions: 16000, Duplicate Protection: On,
      AC Name: pppoe-server-1
```

Alternatively, you can enter `show pppoe underlying-interfaces detail` to display the state and PPPoE family configuration for all configured underlying interfaces.

Meaning

The output shows the name of the underlying interface, the member links of the aggregated bundle, and the PPPoE family configuration. The output shows packet counts when traffic is present on the logical interface.

Verifying the pp0 Interface Configuration

Purpose

Verify that the interface values match your configuration.

Action

From operational mode, enter the `show interfaces pp0` command.

```
user@host> show interfaces pp0.100
  Logical interface pp0.100 (Index 71) (SNMP ifIndex 710)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionUp, Session ID: 1,
      Session AC name: pppoe-server-1, Remote MAC address: 00:00:5e:00:53:34,
      Underlying interface: demux0.100 (Index 70)
    Link:
      ge-5/0/3.32767
      ge-5/1/2.32767
    Input packets : 18572
    Output packets: 18572
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 0 (never), Output: 18566 (00:00:02 ago)
  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
```

Meaning

This output shows information about the PPPoE logical interface created on the underlying VLAN demux interface. The output includes the PPPoE family and aggregated Ethernet redundant link information, and shows input and output traffic for the PPPoE interface.

RELATED DOCUMENTATION

Subscriber Interfaces and Demultiplexing Overview | 93

Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview | 116

Configuring the PPPoE Family for an Underlying Interface | 201

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Static Underlying VLAN Demux Interface over Aggregated Ethernet

IN THIS SECTION

This example shows how you can configure dynamic PPPoE subscriber interfaces over aggregated Ethernet bundles to provide subscriber link redundancy.

Requirements

PPPoE over VLAN demux interfaces over aggregated Ethernet requires the following hardware and software:

- MX Series 5G Universal Routing Platforms
- MPCs
- Junos OS Release 11.2 or later

No special configuration beyond device initialization is required before you can configure this feature.

Overview

Aggregated Ethernet bundles enable link redundancy between the router and networking devices connected by Ethernet links. This example describes how to configure link redundancy for dynamic PPPoE subscribers over aggregated Ethernet interface, `ae0`, with an intermediate static VLAN demux interface, `demux0.100`. Sample tasks include configuring a two-member aggregated Ethernet bundle, configuring a static VLAN demux interface that underlies the PPPoE subscriber interface, and configuring the dynamic profile that establishes the dynamic PPPoE subscriber interfaces.

The dynamic PPPoE profile (`pppoe-profile`) creates the PPPoE subscriber interface. It also configures the router to act as a PPPoE server and enables the local address to be derived from the specified address without assigning an explicit IP address to the interface. The `pppoe-profile` dynamic profile is assigned to the static, intermediate VLAN demux interface (`demux0.100`), which is configured with the PPPoE family (`family pppoe`) attributes. This dynamic profile includes the following predefined variables:

- `$junos-interface-unit`—Represents the logical unit number of the dynamic PPPoE logical interface. This predefined variable is dynamically replaced with the unit number supplied by the router when the subscriber logs in.
- `$junos-underlying-interface`—Represents the name of the underlying Ethernet interface. This predefined variable is dynamically replaced with the interface name supplied by the router when the subscriber logs in.

This example does not show all possible configuration choices.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure link redundancy for dynamic PPPoE subscribers over a static VLAN demux interface over aggregated Ethernet, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
[edit]
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-5/0/3 gigether-options 802.3ad ae0
set interfaces ge-5/0/3 gigether-options 802.3ad primary
set interfaces ge-5/1/2 gigether-options 802.3ad ae0
set interfaces ge-5/1/2 gigether-options 802.3ad backup
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 aggregated-ether-options link-protection
set interfaces demux0 unit 100 vlan-id 100
set interfaces demux0 unit 100 demux-options underlying-interface ae0
set interfaces demux0 unit 100 family pppoe access-concentrator pppoe-server-1
set interfaces demux0 unit 100 family pppoe duplicate-protection
set interfaces demux0 unit 100 family pppoe dynamic-profile pppoe-profile
edit dynamic-profiles pppoe-profile
edit interfaces pp0 unit $junos-interface-unit
set pppoe-options underlying-interface $junos-underlying-interface
```

```
set pppoe-options server
set family inet unnumbered-address lo0.0
top
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure link redundancy for dynamic PPPoE subscribers over a static VLAN demux interface over aggregated Ethernet:

1. Define the number of aggregated Ethernet devices on the router.

   ```
   [edit chassis]
   user@host# set aggregated-devices ethernet device-count 1
   ```

2. Configure a two-link aggregated Ethernet logical interface to serve as the underlying interface for the static VLAN demux subscriber interface. In this example, the LAG bundle is configured for one-to-one active/backup link redundancy. To support link redundancy at the MPC level, the LAG bundle attaches to ports from two different MPCs.

   ```
   [edit interfaces]
   user@host# set ge-5/0/3 gigether-options 802.3ad ae0
   user@host# set ge-5/0/3 gigether-options 802.3ad primary
   user@host# set ge-5/1/2 gigether-options 802.3ad ae0
   user@host# set ge-5/1/2 gigether-options 802.3ad backup
   ```

3. Enable link protection on the aggregated Ethernet logical interface and configure support for single and dual (stacked) VLAN tags.

   ```
   [edit interfaces]
   user@host# set ae0 aggregated-ether-options link-protection
   user@host# set ae0 flexible-vlan-tagging
   ```

4. Configure the VLAN demux interface over the aggregated Ethernet logical interface.

```
[edit interfaces]
user@host# set demux0 unit 100 vlan-id 100
user@host# set demux0 unit 100 demux-options underlying-interface ae0
```

5. Configure the PPPoE family attributes on the VLAN demux interface, including the dynamic profile.

```
[edit interfaces]
user@host# set demux0 unit 100 family pppoe access-concentrator pppoe-server-1
user@host# set demux0 unit 100 family pppoe duplicate-protection
user@host# set demux0 unit 100 family pppoe dynamic-profile pppoe-profile
```

6. Configure the dynamic profile that creates the PPPoE subscriber interfaces.

```
[edit dynamic-profiles pppoe-profile]
user@host# edit interfaces pp0 unit $junos-interface-unit
[edit dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set pppoe-options underlying-interface $junos-underlying-interface
user@host# set pppoe-options server
user@host# set family inet unnumbered-address lo0.0
```

Results

From configuration mode, confirm the aggregated device configuration by entering the `show chassis` command. Confirm the interface configuration by entering the `show interfaces` command. Confirm the dynamic profile configuration by entering the `show dynamic-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis
aggregated-devices {
    ethernet {
        device-count 1;
```

```
    }
}
```

```
[edit]
user@host# show interfaces
    ge-5/0/3 {
        gigether-options {
            802.3ad {
                ae0;
                primary;
            }
        }
    }
    ge-5/1/2 {
        gigether-options {
            802.3ad {
                ae0;
                backup;
            }
        }
    }
    ae0 {
        flexible-vlan-tagging;
        aggregated-ether-options {
            link-protection;
        }
    }
    demux0 {
        unit 100 {
            vlan-id 100;
            demux-options {
                underlying-interface ae0;
            }
            family pppoe {
                access-concentrator pppoe-server-1
                duplicate-protection;
                dynamic-profile pppoe-profile;
            }
```

```
        }
    }
```

```
[edit]
user@host# show dynamic-profiles
pppoe-profile {
    interfaces {
        pp0 {
            unit $junos-interface-unit {
                pppoe-options {
                    underlying-interface $junos-underlying-interface;
                    server;
                }
                family inet {
                    unnumbered-address lo0.0;
                }
            }
        }
    }
}
```

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

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the Aggregated Ethernet Interface Configuration

Purpose

Verify that the interface values match your configuration, the link is up, and traffic is flowing.

Action

From operational mode, enter the `show interfaces redundancy` command.

```
user@host> show interfaces redundancy
Interface  State            Last change  Primary      Secondary    Current status
ae0        On primary                    ge-5/0/3     ge-5/1/2     both up
```

From operational mode, enter the `show interfaces ae0` command.

```
user@host> show interfaces ae0
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 606
  Link-level type: Ethernet, MTU: 1522, Speed: 1Gbps, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 1, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:00:5e:00:53:d0, Hardware address: 00:00:5e:00:53:d0
  Last flapped   : 2011-03-11 13:24:18 PST (2d 03:34 ago)
  Input rate     : 1984 bps (2 pps)
  Output rate    : 0 bps (0 pps)

  Logical interface ae0.32767 (Index 69) (SNMP ifIndex 709)
    Flags: SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :        371259          2      46036116         1984
        Output:             0          0             0            0
    Protocol multiservice, MTU: Unlimited
      Flags: Is-Primary
```

Meaning

The `show interfaces redundancy` output shows the redundant link configuration and that both link interfaces are up. The `show interfaces ae0` output shows that the aggregated Ethernet interface is up and that traffic is being received on the logical interface.

Verifying the demux0 Interface Configuration

Purpose

Verify that the VLAN demux interface displays the configured PPPoE family attributes and the member links in the aggregated Ethernet bundle.

Action

From operational mode, enter the `show interfaces demux0` command.

```
user@host> show interfaces demux0.100
Logical interface demux0.100 (Index 76) (SNMP ifIndex 61160)
    Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 ]
    Encapsulation: ENET2
    Demux:
      Underlying interface: ae0 (Index 199)
    Link:
      ge-5/0/3
      ge-5/1/2
    Input packets : 2
    Output packets: 18575
    Protocol pppoe
      Dynamic Profile: pppoe-profile,
      Service Name Table: None,
      Max Sessions: 16000, Duplicate Protection: On,
      AC Name: pppoe-server-1
```

Alternatively, you can enter `show pppoe underlying-interfaces detail` to display the state and PPPoE family configuration for all configured underlying interfaces. The output also provides information about PPPoE negotiation on a per-VLAN basis.

Meaning

The output shows the name of the underlying interface, the member links of the aggregated bundle, and the PPPoE family configuration. The output shows packet counts when traffic is present on the logical interface.

RELATED DOCUMENTATION

Example: Configuring a Dynamic PPPoE Subscriber Interface on a Dynamic Underlying VLAN Demux Interface over Aggregated Ethernet

IN THIS SECTION

This example shows how you can configure dynamic PPPoE subscriber interfaces over aggregated Ethernet bundles to provide subscriber link redundancy.

Requirements

PPPoE over VLAN demux interfaces over aggregated Ethernet requires the following hardware and software:

- MX Series 5G Universal Routing Platforms
- MPCs
- Junos OS Release 11.2 or later

No special configuration beyond device initialization is required before you can configure this feature.

Overview

Aggregated Ethernet bundles enable link redundancy between the router and networking devices connected by Ethernet links. This example describes how to configure link redundancy for dynamic

PPPoE subscribers over aggregated Ethernet with an intermediate dynamic VLAN demux interface. Sample tasks include configuring a two-member aggregated Ethernet bundle, configuring dynamic profiles that establish the dynamic VLAN demux interface that underlies the PPPoE subscriber interface, and configuring the dynamic profile that establishes the dynamic PPPoE subscriber interfaces.

In this example, two different dynamic profiles are configured to instantiate either VLAN (`vlan-profile`) or S-VLAN (`svlan-profile`) demux interfaces. These profiles define PPPoE family options and include the dynamic PPPoE profile (`pppoe-profile`) that creates the PPPoE subscriber interface. Junos OS predefined variables are used in each profile to represent the interfaces and VLAN identifiers that are dynamically created. These dynamic profiles include the following predefined variables:

- `$junos-interface-unit`—Represents the logical unit number of the dynamic VLAN demux interface. This predefined variable is dynamically replaced with the unit number supplied by the router when the subscriber logs in.
- `$junos-interface-ifd-name`—Represents the underlying logical interface on which the PPPoE subscriber interface is created. This predefined variable is dynamically replaced with the name of the underlying interface supplied by the router when the subscriber logs in.
- `$junos-vlan-id`—Represents the VLAN identifier. This predefined variable is dynamically replaced with a VLAN ID when the subscriber logs in. The VLAN ID is allocated within the VLAN range specified in the aggregated Ethernet configuration. In the case of the S-VLAN demux, `$junos-vlan-id` represents the inner VLAN identifier.
- `$junos-stacked-vlan-id`—Represents the outer VLAN identifier for the stacked VLAN. This predefined variable is dynamically replaced with a VLAN ID when the subscriber logs in. The VLAN ID is allocated within the VLAN range specified in the aggregated Ethernet configuration. This variable is not used for the VLAN demux configuration.

The dynamic PPPoE profile (`pppoe-profile`) creates the PPPoE subscriber interface. It also configures the router to act as a PPPoE server and enables the local address to be derived from the specified address without assigning an explicit IP address to the interface. The `pppoe-profile` dynamic profile is assigned to the dynamic, intermediate VLAN and S-VLAN demux interfaces. This dynamic profile includes the following predefined variables:

- `$junos-interface-unit`—Represents the logical unit number of the dynamic PPPoE logical interface. This predefined variable is dynamically replaced with the unit number supplied by the router when the subscriber logs in.
- `$junos-underlying-interface`—Represents the name of the underlying Ethernet interface. This predefined variable is dynamically replaced with the interface name supplied by the router when the subscriber logs in.

This example does not show all possible configuration choices.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure link redundancy for dynamic PPPoE subscribers over a dynamic VLAN demux interface over aggregated Ethernet, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
[edit]
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-5/0/3 gigether-options 802.3ad ae0
set interfaces ge-5/0/3 gigether-options 802.3ad primary
set interfaces ge-5/1/2 gigether-options 802.3ad ae0
set interfaces ge-5/1/2 gigether-options 802.3ad backup
edit interfaces ae0
set flexible-vlan-tagging
set aggregated-ether-options link-protection
edit auto-configure
set vlan-ranges dynamic-profile vlan-profile accept pppoe
set vlan-ranges dynamic-profile vlan-profile ranges 1-4094
set stacked-vlan-ranges dynamic-profile svlan-profile accept pppoe
set stacked-vlan-ranges dynamic-profile svlan-profile ranges 1-4094,1-4094
top
edit dynamic-profiles pppoe-profile
edit interfaces pp0 unit $junos-interface-unit
set pppoe-options underlying-interface $junos-underlying-interface
set pppoe-options server
set family inet unnumbered-address lo0.0
top
edit dynamic-profiles vlan-profile interfaces demux0
edit unit $junos-interface-unit
set vlan-id $junos-vlan-id
set demux-options underlying-interface $junos-interface-ifd-name
```

```
set family pppoe access-concentrator pppoe-server-1
set family pppoe duplicate-protection
set family pppoe dynamic-profile pppoe-profile
top
edit dynamic-profiles svlan-profile interfaces demux0
edit unit $junos-interface-unit
set vlan-tags outer $junos-stacked-vlan-id
set vlan-tags inner $junos-vlan-id
set demux-options underlying-interface $junos-interface-ifd-name
set family pppoe access-concentrator pppoe-server-1
set family pppoe duplicate-protection
set family pppoe dynamic-profile pppoe-profile
top
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure link redundancy for dynamic PPPoE subscribers over a dynamic VLAN demux interface over aggregated Ethernet:

1. Define the number of aggregated Ethernet devices on the router.

```
[edit chassis]
user@host# set aggregated-devices ethernet device-count 1
```

2. Configure a two-link aggregated Ethernet logical interface to serve as the underlying interface for the dynamic VLAN demux subscriber interface. In this example, the LAG bundle is configured for one-to-one active/backup link redundancy. To support link redundancy at the MPC level, the LAG bundle attaches to ports from two different MPCs.

```
[edit interfaces]
user@host# set ge-5/0/3 gigether-options 802.3ad ae0
user@host# set ge-5/0/3 gigether-options 802.3ad primary
user@host# set ge-5/1/2 gigether-options 802.3ad ae0
user@host# set ge-5/1/2 gigether-options 802.3ad backup
```

3. Enable link protection on the aggregated Ethernet logical interface and configure support for single and dual (stacked) VLAN tags.

```
[edit interfaces]
user@host# set ae0 aggregated-ether-options link-protection
user@host# set ae0 flexible-vlan-tagging
```

4. Configure the parameters for automatically configuring VLANs and S-VLANs, including the VLAN ranges and dynamic profiles.

```
[edit interfaces]
user@host# set ae0 auto-configure vlan-ranges dynamic-profile vlan-profile accept pppoe
user@host# set ae0 auto-configure vlan-ranges dynamic-profile vlan-profile ranges 1-4094
user@host# set ae0 auto-configure stacked-vlan-ranges dynamic-profile svlan-profile accept
pppoe
user@host# set ae0 auto-configure stacked-vlan-ranges dynamic-profile svlan-profile ranges
1-4094,1-4094
```

5. Configure the dynamic profile that creates the PPPoE subscriber interface.

```
[edit dynamic-profiles pppoe-profile]
user@host# edit interfaces pp0 unit $junos-interface-unit
[edit dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set pppoe-options underlying-interface $junos-underlying-interface
user@host# set pppoe-options server
user@host# set family inet unnumbered-address lo0.0
```

6. Configure the dynamic profile that creates VLAN demux underlying interfaces, including the PPPoE family attributes.

```
[edit dynamic-profiles vlan-profile]
user@host# edit interfaces demux0 unit $junos-interface-unit
[edit dynamic-profiles vlan-profile interfaces demux0 unit "$junos-interface-unit"]
user@host# set vlan-id $junos-vlan-id
user@host# set demux-options underlying-interface $junos-interface-ifd-name
user@host# set family pppoe access-concentrator pppoe-server-1
user@host# set family pppoe duplicate-protection
user@host# set family pppoe dynamic-profile pppoe-profile
```

7. Configure the dynamic profile that creates S-VLAN demux underlying interfaces, including the PPPoE family attributes.

```
[edit dynamic-profiles svlan-profile]
user@host# edit interfaces demux0 unit $junos-interface-unit
[edit dynamic-profiles svlan-profile interfaces demux0 unit "$junos-interface-unit"]
user@host# set vlan-tags outer $junos-stacked-vlan-id
user@host# set vlan-tags inner $junos-vlan-id
user@host# set demux-options underlying-interface $junos-interface-ifd-name
user@host# set family pppoe access-concentrator pppoe-server-1
user@host# set family pppoe duplicate-protection
user@host# set family pppoe dynamic-profile pppoe-profile
```

Results

From configuration mode, confirm the aggregated device configuration by entering the `show chassis` command. Confirm the interface configuration by entering the `show interfaces` command. Confirm the dynamic profile configuration by entering the `show dynamic-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis
aggregated-devices {
    ethernet {
        device-count 1;
    }
}
```

```
[edit]
user@host# show interfaces
    ge-5/0/3 {
        gigether-options {
            802.3ad {
                ae0;
                primary;
            }
        }
    }
    ge-5/1/2 {
```

```
        gigether-options {
            802.3ad {
                ae0;
                backup;
            }
        }
    }
    ae0 {
        flexible-vlan-tagging;
        aggregated-ether-options {
            link-protection;
        }
        auto-configure {
            vlan-ranges {
                dynamic-profile {
                    vlan-profile {
                        accept pppoe;
                        vlan-ranges 1-4094};
                    }
                }
            }
            stacked-vlan-ranges {
                dynamic-profile {
                    svlan-profile {
                        accept pppoe;
                        vlan-ranges 1-4094,1-4094;
                    }
                }
            }
        }
    }
```

```
[edit]
user@host# show dynamic-profiles
pppoe-profile {
    interfaces {
        pp0 {
            unit $junos-interface-unit {
                pppoe-options {
                    underlying-interface $junos-underlying-interface;
                    server;
```

```
                }
                family inet {
                    unnumbered-address lo0.0;
                }
            }
        }
    }
}
vlan-profile {
    interfaces {
        demux0 {
            unit "$junos-interface-unit" {
                vlan-id "$junos-vlan-id";
                demux-options {
                    underlying-interface "$junos-interface-ifd-name";
                }
                family pppoe {
                    access-concentrator pppoe-server-1;
                    duplicate-protection;
                    dynamic-profile pppoe-profile;
                }
            }
        }
    }
}
svlan-profile {
    interfaces {
        demux0 {
            unit "$junos-interface-unit" {
                vlan-tags outer "$junos-stacked-vlan-id" inner "$junos-vlan-id";
                demux-options {
                    underlying-interface "$junos-interface-ifd-name";
                }
                family pppoe {
                    access-concentrator pppoe-server-1;
                    duplicate-protection;
                    dynamic-profile pppoe-profile;
                }
            }
        }
    }
}
```

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

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform this task:

Verifying the Aggregated Ethernet Interface Configuration

Purpose

Verify that the interface values match your configuration, the link is up, and traffic is flowing.

Action

From operational mode, enter the `show interfaces redundancy` command.

```
user@host> show interfaces redundancy
Interface  State            Last change  Primary     Secondary    Current status
ae0        On primary                    ge-5/0/3    ge-5/1/2     both up
```

From operational mode, enter the `show interfaces ae0` command.

```
user@host> show interfaces ae0
Physical interface: ae0, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 606
  Link-level type: Ethernet, MTU: 1522, Speed: 1Gbps, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 1, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 00:00:5e:00:53:d0, Hardware address: 00:00:5e:00:53:d0
  Last flapped   : 2011-03-11 13:24:18 PST (2d 03:34 ago)
  Input rate     : 1984 bps (2 pps)
```

```
Output rate    : 0 bps (0 pps)

Logical interface ae0.32767 (Index 69) (SNMP ifIndex 709)
  Flags: SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
  Statistics        Packets        pps         Bytes          bps
  Bundle:
      Input :        371259          2      46036116         1984
      Output:             0          0             0            0
  Protocol multiservice, MTU: Unlimited
    Flags: Is-Primary
```

Meaning

The `show interfaces redundancy` output shows the redundant link configuration and that both link interfaces are up. The `show interfaces ae0` output shows that the aggregated Ethernet interface is up and that traffic is being received on the logical interface.

RELATED DOCUMENTATION

CHAPTER 18

Configuring PPPoE Session Limits

IN THIS CHAPTER

PPPoE Maximum Session Limit Overview

IN THIS SECTION

The maximum session limit for PPPoE subscriber interfaces specifies the maximum number of concurrent static or dynamic PPPoE logical interfaces (sessions) that the router can activate on the PPPoE underlying interface, or the maximum number of active static or dynamic PPPoE sessions that the router can establish with a particular service entry in a PPPoE service name table.

You can configure the PPPoE maximum session limit in one of two ways:

- On a per-interface basis.
- (Default) On a per-subscriber basis.

This overview describes the concepts you need to understand to configure the PPPoE maximum session limit, and covers the following topics:

Per-Interface Configuration for PPPoE Maximum Session Limit Using the CLI

When you configure the PPPoE maximum session limit for a particular interface, you can use the `max-sessions` statement to specify either or both of the following:

- The maximum number of concurrent PPPoE sessions that the router can activate on the PPPoE underlying interface
- The maximum number of active PPPoE sessions using either static or dynamic PPPoE interfaces that the router can establish with a particular named service entry, `empty` service entry, or `any` service entry in a PPPoE service name table

You can configure the PPPoE maximum session value from 1 through the platform-specific default for your router. The default value is equal to the maximum number of PPPoE sessions supported on your routing platform. If the number of active PPPoE sessions exceeds the value configured, the router prohibits creation of any new PPPoE sessions, and the PPPoE application on the router returns a PPPoE Active Discovery Session (PADS) packet with an error to the PPPoE client.

Changing the PPPoE maximum session value has no effect on dynamic PPPoE subscriber interfaces that are already active.

Per-Subscriber Configuration for PPPoE Maximum Session Limit Using RADIUS

To configure the PPPoE maximum session limit for a particular subscriber, you can use the value returned by the RADIUS server in the Max-Clients-Per-Interface Juniper Networks VSA [26-143] during the subscriber authentication process. For PPPoE clients, the Max-Clients-Per-Interface VSA returns the maximum number of sessions (PPPoE subinterfaces) per PPPoE major interface.

By default, the PPPoE maximum session value returned by RADIUS in the Max-Clients-Per-Interface VSA takes precedence over the PPPoE maximum session value configured with the `max-sessions` statement.

If you configure multiple subscribers on the same PPPoE underlying VLAN interface and RADIUS returns a different PPPoE maximum session value for each subscriber, the router uses the most recent PPPoE maximum session value returned by RADIUS to determine whether to override the current PPPoE maximum session value and create the new PPPoE session.

The following sequence describes how the router obtains the PPPoE maximum session value from RADIUS when a PPPoE subscriber logs in to initiate a session with the router. (In a PPPoE subscriber network, the router functions as a *remote access concentrator*, also known as a *PPPoE server*.)

1. The PPPoE client and the router participate in the PPPoE Discovery process to establish the PPPoE connection.
2. The PPP Link Control Protocol (LCP) negotiates the PPP link between the client and the router.

3. The PPP application sends the subscriber authentication request to the AAA application.

4. AAA sends the authentication request to an external RADIUS server.

5. The RADIUS server returns the PPPoE maximum session value for that subscriber to AAA in the Max-Clients-Per-Interface VSA as part of an Access-Accept message.

   The RADIUS server does not return the Max-Clients-Per-Interface VSA in Change of Authorization Request (CoA-Request) messages.

6. AAA passes the response from RADIUS to PPP.

7. PPP validates the subscriber parameters and, if authentication succeeds, passes the PPPoE maximum session value returned by RADIUS to the PPPoE application.

8. PPPoE uses the maximum session value returned by RADIUS to determine whether to override the current PPPoE maximum session value and create or tear down the new PPPoE session.

Override of PPPoE Maximum Session Limit from RADIUS

You can configure the router to ignore (clear) the PPPoE maximum session value returned by the RADIUS server in the Max-Clients-Per-Interface VSA. Configuring the router to ignore the VSA restores the PPPoE maximum session value on the underlying interface to the value configured in the CLI.

Session Limit Enforcement Using Circuit-ID and MAC address for PPPoE Subscribers

You can manage PPPoE subscriber sessions using Circuit-ID and MAC address information for session limit enforcement. This feature ensures session management by enforcing session limits even when ACI or ARI tags are unavailable. This feature combines Circuit-ID and **source MAC address**, for MAC address-based interface sets when agent circuit identifier (ACI) and agent remote identifier (ARI) tags are unavailable. **The RG MAC address is replaced by a unique source MAC Address by the Access Node (AN).** This feature allows service providers to efficiently manage and control the number of PPPoE sessions per household, even in diverse access node environments. Dynamic VLAN support ensures flexible and scalable subscriber session management, while high availability is maintained with Graceful Routing Engine Switchover (GRES) and in-service software upgrade (ISSU). By utilizing MAC address-triggered ACI VLANs, service providers can enforce session limits effectively, for consistent management across various types of network nodes. The MAC address line identity option is activated when no line identities are received and ACI VLANs are created based on MAC address received. See Standard and Vendor-Specific RADIUS Attributes.

RELATED DOCUMENTATION

Guidelines for Using PPPoE Maximum Session Limit from RADIUS | 236
Limiting the Maximum Number of PPPoE Sessions on the Underlying Interface | 237
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Example: Concurrent Configuration of Dynamic DHCP IP Demux and PPPoE Demux Interfaces over the Same VLAN Demux Interface | 151

Guidelines for Using PPPoE Maximum Session Limit from RADIUS

Consider the following guidelines when you use the PPPoE maximum session value returned by RADIUS in the Max-Clients-Per-Interface vendor-specific attribute (VSA) [26-143]:

- If the current number of sessions (including newly created sessions) is *less than* the new PPPoE maximum session value returned by RADIUS, the PPPoE application overrides the current value and enables interface creation to proceed.
- If the current number of sessions (including newly created sessions) is *equal to* the new PPPoE maximum session value returned by RADIUS, the PPPoE application overrides the current value and enables interface creation to proceed.
- If the current number of sessions (including newly created sessions) is *greater than* the new PPPoE maximum session value returned by RADIUS, the PPPoE application overrides the current value and brings down the new interface.

To illustrate these guidelines, Table 10 on page 236 shows examples of how the router handles the PPPoE session when the current number of sessions is less than (first row), equal to (second row), and greater than (third row) the new PPPoE maximum session value returned by RADIUS when a new subscriber logs in.

Table 10: Sample PPPoE Maximum Session Values During Subscriber Login

New PPPoE Maximum Session Value from RADIUS	Current PPPoE Maximum Session Value	Existing Number of PPPoE Sessions	New PPPoE Maximum Session Value	New Number of PPPoE Sessions	Status of Session
10	5	4	10	5	PPPoE session up
5	5	4	5	5	PPPoE session up
3	5	4	3	4	PPPoE session down

RELATED DOCUMENTATION

PPPoE Maximum Session Limit Overview | 233

Juniper Networks VSAs Supported by the AAA Service Framework

Limiting the Maximum Number of PPPoE Sessions on the Underlying Interface | 237

Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Limiting the Maximum Number of PPPoE Sessions on the Underlying Interface

You can limit the number of concurrent static or dynamic PPPoE logical interfaces (sessions) that the router can activate on the PPPoE underlying interface, or the number of active static or dynamic PPPoE sessions that the router can establish with a particular service entry in a PPPoE service name table.

To configure the PPPoE maximum session limit:

1. Specify that you want to configure PPPoE-specific options on the underlying interface:

   - For a PPPoE family in a dynamic profile for a VLAN demultiplexing (demux) logical interface:

     ```
     [edit dynamic-profiles profile-name interfaces demux0 unit logical-unit-number]
     user@host# edit family pppoe
     ```

   - For a PPPoE family in a dynamic profile:

     ```
     [edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
     user@host# edit family pppoe
     ```

   - For a PPPoE underlying interface in a dynamic profile:

     ```
     [edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
     user@host# edit pppoe-underlying-options
     ```

   - For a PPPoE family on an underlying interface:

     ```
     [edit interfaces interface-name unit logical-unit-number]
     user@host# edit family pppoe
     ```

- For an underlying interface with PPPoE encapsulation:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit pppoe-underlying-options
```

- For an underlying interface established with a particular service entry in a PPPoE service name table:

```
[edit protocols pppoe service-name-tables table-name]
user@host# edit service service-name
```

2. Configure the maximum number of concurrent PPPoE sessions that the router can activate on the underlying interface in either of the following ways:

   - To configure the maximum number of concurrent PPPoE sessions on a per-interface basis, from 1 to the platform-specific default for your router, use the `max-sessions` statement:

   ```
   [edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
   user@host# set max-sessions number
   ```

   - To configure the maximum number of concurrent PPPoE sessions on a per-subscriber basis, use the value returned by RADIUS in the Max-Clients-Per-Interface Juniper Networks vendor-specific attribute (VSA) [26-143]. By default, the PPPoE maximum session value returned by RADIUS in the Max-Clients-Per-Interface VSA takes precedence over the PPPoE maximum session value configured with the `max-sessions` statement.

3. (Optional) To restore the PPPoE maximum session value on the underlying interface to the value configured in the CLI with the `max-sessions` statement, configure the router to ignore the value returned by RADIUS in the Max-Clients-Per-Interface VSA.

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set max-sessions-vsa-ignore
```

NOTE: You can issue the `max-sessions-vsa-ignore` statement at the same hierarchy levels as the `max-sessions` statement, with the exception of the `[edit protocols pppoe service-name-tables *table-name* service *service-name*]` hierarchy level.

RELATED DOCUMENTATION

PPPoE Maximum Session Limit Overview | 233
Guidelines for Using PPPoE Maximum Session Limit from RADIUS | 236
Juniper Networks VSAs Supported by the AAA Service Framework
Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198
Configuring the PPPoE Family for an Underlying Interface | 201
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

CHAPTER 19

Configuring PPPoE Subscriber Session Lockout

IN THIS CHAPTER

PPPoE Subscriber Session Lockout Overview

IN THIS SECTION

PPPoE subscriber session lockout, also called *PPPoE encapsulation type lockout*, temporarily prevents (locks out) a failed or short-lived static or dynamic PPPoE subscriber session from reconnecting for a certain period of time. This time period, known as the *lockout period*, is derived from a formula and increases exponentially based on the number of successive reconnection failures.

You can configure PPPoE subscriber session lockout, also known as *short-cycle protection*, for VLAN, VLAN demultiplexing (demux), and PPP-over-Ethernet-over-ATM (PPPoE-over-ATM) dynamic subscriber interfaces.

This overview describes the concepts you need to understand to configure PPPoE subscriber session lockout, and covers the following topics:

Benefits of Using PPPoE Subscriber Session Lockout

PPPoE subscriber session lockout provides the following benefits:

- Reduces excessive loading on the router by:
  - Reducing the resources required to process PPPoE control packets to negotiate and terminate short-lived connections
  - Reducing the resources required to allocate and deallocate services, such as *class of service* (CoS) and firewall filters, for failed or short-lived subscriber sessions
  - Temporarily deferring failed or short-lived subscriber sessions in favor of sessions that can complete successfully.
- Reduces excessive loading on external authentication, authorization, and accounting (AAA) servers, such as RADIUS or Diameter:
  - As a result of failed or short-lived PPPoE subscriber sessions that occur repeatedly for the same subscriber
  - By reducing the resources required to authenticate and terminate these connections
- Enables lockout of a single failed or short-lived PPP session without disrupting other PPP sessions on the same PPPoE underlying interface

  Because PPPoE subscriber session lockout identifies each subscriber session by either its unique media access control (MAC) source address on the underlying interface or by its agent circuit identifier (ACI) value, the router can lock out only the offending PPP session while enabling other PPP sessions on the same underlying interface to successfully negotiate the connection.

Conditions That Cause Short-Lived PPPoE Subscriber Sessions

Conditions that can cause a short-lived subscriber session include:

- Authentication denials from external AAA servers, such as RADIUS, due to the absence of a corresponding entry in the RADIUS database or due to improper login attempts
- Configuration errors within a dynamic profile or RADIUS record

- Insufficient memory resources to create a dynamic PPPoE subscriber interface
- Protocol failure or error within the dynamic PPPoE subscriber interface
- Client logout shortly after a successful login; this action creates a complete dynamic PPPoE subscriber interface before the interface is torn down

How PPPoE Subscriber Session Lockout Works

PPPoE subscriber session lockout is disabled on the router by default. When you enable PPPoE subscriber session lockout, the router does the following:

1. Detects a short-lived subscriber session, also referred to as a *short-cycle event*.

   A short-lived subscriber session is detected, partially or completely created, and terminated by the router within 150 seconds. The router identifies each PPPoE subscriber session by its unique MAC source address on the PPPoE underlying interface or by its ACI value.

2. Tracks the time between repeated short-cycle events to determine whether to increase the lockout time for a subsequent short-cycle event.

3. Applies a time penalty for each short-cycle event based on a default or configured lockout period and the number of consecutive short-cycle events that occur repeatedly for the same subscriber.

4. Temporarily locks out the specified PPPoE subscriber by preventing connection to the router.

   During lockout, the router drops negotiation packets for the PPPoE subscriber session until the lockout period expires. When the lockout period expires, the PPPoE subscriber session and its associated MAC source address or ACI value resume normal negotiation of the connection.

PPPoE Subscriber Session Lockout on ACI-Based Interfaces

By default, the router identifies a subscriber session using the unique MAC source address on the PPPoE underlying interface. You can configure subscriber session lockout based on the ACI string of the underlying interface, which allows you to lock out all PPPoE subscriber sessions from the same household.

The ACI string is contained in the DSL Forum Agent-Circuit-ID VSA [26-1] (option 0x105) of PPPoE Active Discovery Initiation (PADI) and PPPoE Active Discovery Request (PADR) control packets. This option locks out all PPPoE subscriber sessions on the underlying interface that share the same ACI string in their PPPoE PADI and PADR control packets.

PPPoE subscriber session lockout based on the ACI value is useful when MAC source addresses are not unique on the PPPoE underlying interface. For example:

- PPPoE interworking function sessions in which the MAC addresses of all PPPoE inter-working function sessions contain the MAC address of the DSLAM device
- Configurations in which the access node (usually a DSLAM device) overwrites the MAC source address in PPPoE packets received from the customer premises equipment (CPE) with its own MAC address for security purposes
- Duplicate MAC source addresses across disparate households in an N:1 (service VLAN) configuration, which requires the router to use a combination of the MAC source address and the ACI value to uniquely identify a subscriber

PPPoE Subscriber Session Lockout and Duplicate Protection

Duplicate protection, which is disabled on the router by default, prevents the activation of another PPPoE subscriber session on the same PPPoE underlying interface when a PPPoE subscriber session with the same media access control (MAC) address is already active on that interface. When you configure PPPoE subscriber session lockout, we recommend that you enable duplicate protection to ensure that the MAC source address for each active PPPoE session is unique on the underlying interface.

With PPPoE subscriber session lockout configured, the router identifies subscriber sessions by their unique MAC source address. If the router detects a short-lived (short-cycle) subscriber session, it applies the default or configured lockout period to that MAC source address to temporarily prevent reconnection. If the MAC source address is not unique on the underlying interface, multiple PPPoE subscriber sessions with the same MAC source address might also be affected by the lockout.

Persistence of the Lockout Condition After Automatic Removal of Dynamic Subscriber VLANs

You can configure automatic removal of subscriber VLANs that have no PPPoE client sessions by issuing the `remove-when-no-subscribers` statement at the `[edit interfaces` *`interface-name`* `auto-configure]` hierarchy level. If PPPoE subscriber session lockout is also configured on the interface, the lockout condition persists even after the router has removed the dynamic VLAN or VLAN demux subscriber interface.

When you configure both PPPoE subscriber session lockout and automatic removal of subscriber VLANs with no client sessions, the lockout condition for the affected subscriber sessions persists until the lockout timer expires for each PPPoE client undergoing lockout on the underlying interface. If you create the VLAN or VLAN demux subscriber interface again before all timers expire, the lockout condition persists for the newly created subscriber interface.

Use of Encapsulation Type Identifiers to Clear or Display the Lockout Condition

You can clear the lockout condition for a specific MAC source address or ACI value, all MAC source addresses or ACI values, or for an ACI value that matches a UNIX-based regular expression by specifying VLAN or ATM encapsulation type identifier options in the `clear pppoe lockout vlan-identifier` or `clear pppoe lockout atm-identifier` command, respectively. Similarly, you can display information about the lockout condition and the status of affected subscriber sessions by including encapsulation type identifier options in the `show pppoe lockout vlan-identifier` or `show pppoe lockout atm-identifier` command. Specifying encapsulation type lockout identifiers enables you to clear or display the lockout condition when no underlying interface exists for the subscriber session.

For the VLAN encapsulation type on VLAN and VLAN demux subscriber interfaces, the identifier options include:

- Device name (physical interface or aggregated Ethernet bundle)
- S-VLAN ID (outer tag)
- VLAN ID (inner tag)

For the ATM encapsulation type on PPPoE-over-ATM subscriber interfaces, the identifier options include:

- Device name (physical interface or aggregated Ethernet bundle)
- Virtual path identifier (VPI)
- Virtual circuit identifier (VCI)

Termination of the Lockout Condition

When a PPPoE subscriber session identified by either an ACI value or a unique MAC source address is undergoing lockout, the lockout condition persists until all lockout timers have expired, *except* when either of the following occurs:

- You administratively clear the lockout condition by issuing the `clear pppoe lockout` operational command.
- You reset the interface module on which the subscriber session undergoing lockout is configured.

When you clear the lockout condition or reset the interface module, the router terminates lockout for all PPPoE subscriber sessions on the underlying interface, and clears the lockout history for all affected subscriber sessions.

RELATED DOCUMENTATION

Understanding the Lockout Period for PPPoE Subscriber Session Lockout

IN THIS SECTION

When you configure PPPoE subscriber session lockout, the router applies a time penalty called the *lockout period* for each failed or short-lived subscriber session.

This overview describes how the router determines and applies the PPPoE subscriber session lockout period, and covers the following topics:

Duration of PPPoE Subscriber Session Lockout Period

The duration of the lockout period is based on a default or configured lockout time and the number of consecutive short-cycle (short-lived) events that occur repeatedly for the same subscriber. When you include the `short-cycle-protection` statement to configure PPPoE subscriber session lockout on a PPPoE underlying interface, you can use the default lockout time range of 1 through 300 seconds (5 minutes), or you can override the default lockout period by configuring a nondefault lockout time in the range 1 through 86,400 seconds (24 hours).

The lockout time penalty applied by the router for each short-cycle event differs depending on the event. For example, some short-cycle events represent normal subscriber behavior, such as a PPPoE subscriber logging in once per hour to check e-mail and logging out shortly thereafter. The router does not noticeably penalize a subscriber for these types of events.

By contrast, other short-cycle events are the result of repeated attempts to log in to the router for reasons such as an incorrectly typed password, customer premises equipment (CPE) that performs repeated auto-retries, or malicious attempts to access the Internet illegally. For these types of short-cycle events, the router applies a lockout time penalty that starts with a short time interval and increases exponentially. In these instances, the initial lockout time is short enough to avoid noticeably penalizing a subscriber who, for example, types a password incorrectly several times before entering the correct one.

For example, using the default lockout time range of 1 through 300 seconds, the increasing lockout period on the router is: 1 second, 2 seconds, 4 seconds, 8 seconds, 16 seconds, 32 seconds, 64 seconds, 128 seconds, 256 seconds, and finally, 300 seconds (5 minutes).

How the Router Determines the PPPoE Subscriber Session Lockout Period

The router uses the following rules to determine the PPPoE subscriber session lockout period for short-lived PPPoE subscriber sessions:

- The lockout period is derived from the following formula:

  (minimum lockout time) * (2 ^ *n*-1)

  where *n* represents the number of consecutive short-cycle events for the same subscriber. The router identifies a PPPoE subscriber session by its MAC source address, which should be unique on the underlying PPPoE interface, or ACI value.

- The router increments the value of *n* when the time between short-cycle events is either within 15 minutes or the maximum lockout time, whichever is greater.

- When the time between short-cycle events is greater than either 15 minutes or the maximum lockout time, the value of *n* reverts to 1. This condition is referred to as a *lockout grace period*.

- The lockout period never exceeds the maximum configured lockout time.

  For example, for a configured (nondefault) lockout time in the range 20 through 120 seconds, the increasing lockout period on the router is: 20 seconds, 40 seconds, 80 seconds, and finally, 120 seconds (2 minutes).

- A *short-cycle event* is detected, partially or completely created, and terminated by the router within 150 seconds. The router tracks the time between short-cycle events to determine whether to increase the lockout time for a subsequent short-cycle event for the same subscriber.

  > **NOTE**: When the calculated lockout time is equal to or exceeds the maximum lockout time, the router uses the maximum lockout time value until the time to the next short-cycle event exceeds the greater of 15 minutes or the maximum lockout time value. At that point, the lockout time reverts to the minimum lockout time value.

- The minimum lockout time value cannot exceed the maximum lockout time value.

When the minimum and maximum lockout time values are equal, the lockout time becomes fixed at that value.

RELATED DOCUMENTATION

PPPoE Subscriber Session Lockout Overview | 240
Configuring Lockout of PPPoE Subscriber Sessions | 247
Clearing Lockout of PPPoE Subscriber Sessions | 250
Verifying and Managing Dynamic PPPoE Configuration | 292
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Configuring Lockout of PPPoE Subscriber Sessions

You can configure the router to temporarily lock out a failed or short-lived PPPoE subscriber session from reconnecting for a period of time. The PPPoE subscriber session can reside on a VLAN, VLAN demux, or PPPoE-over-ATM underlying interface.

Before you begin:

- Configure the PPPoE underlying interface.

  To configure the underlying interface for use with a PPPoE dynamic profile, see "Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces" on page 198.

  To configure the PPPoE family for an underlying interface, see "Configuring the PPPoE Family for an Underlying Interface" on page 201.

To configure temporary lockout of PPPoE subscriber sessions:

1. Specify that you want to configure PPPoE-specific options on the underlying interface:

   - For a PPPoE family in a dynamic profile for a VLAN demultiplexing (demux) logical interface:

     ```
     [edit dynamic-profiles profile-name interfaces demux0 unit logical-unit-number]
     user@host# edit family pppoe
     ```

- For a PPPoE family in a dynamic profile:

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
user@host# edit family pppoe
```

- For a PPPoE underlying interface in a dynamic profile:

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number]
user@host# edit pppoe-underlying-options
```

- For a PPPoE family on an underlying interface:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit family pppoe
```

- For an underlying interface with PPPoE encapsulation:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit pppoe-underlying-options
```

- For a PPPoE family in a dynamic profile for a PPPoE-over-ATM logical interface:

```
[edit dynamic-profiles profile-name interfaces at-fpc/pic/port unit logical-unit-number]
user@host# edit family pppoe
```

- For a PPPoE family on an underlying ATM logical interface:

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# edit family pppoe
```

2. Enable duplicate protection to prevent negotiation of a dynamic or static PPPoE client session on the same underlying interface when a PPPoE client session with the same media access control (MAC) source address is already active on that interface.

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set duplicate-protection
```

BEST PRACTICE: When you configure PPPoE subscriber session lockout, we recommend that you enable duplicate protection to ensure that the MAC source address for each PPPoE session is unique on the underlying interface.

3. Enable PPPoE subscriber session lockout using one of the following filtering mechanisms to identify the subscriber sessions for lockout:

   - Media access control (MAC)-address based subscriber session lockout (default)

     - To configure MAC-based subscriber session lockout with the default lockout period of 1 through 300 seconds:

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set short-cycle-protection
```

     - To configure MAC-based subscriber session lockout with a nondefault lockout period:

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set short-cycle-protection lockout-time-min minimum-seconds lockout-time-
max maximum-seconds
```

   - Agent circuit identifier (ACI)-based subscriber session lockout

     - To configure ACI-based subscriber session lockout with the default lockout period:

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set short-cycle-protection filter aci
```

For example, the following statement configures temporary lockout based on ACI information for subscriber sessions on a dynamic VLAN demux underlying interface. It uses the default lockout time range 1 through 300 seconds.

```
[edit dynamic-profiles my-demux-vlan-profile interfaces demux0 unit “$junos-interface-
unit” family pppoe]
user@host# set short-cycle-protection filter aci
```

- To configure ACI-based subscriber session lockout with a nondefault lockout period:

```
[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]
user@host# set short-cycle-protection lockout-time-min minimum-seconds lockout-time-
max maximum-seconds filter aci
```

For example, the following statement configures temporary lockout based on ACI information for subscriber sessions on a dynamic VLAN underlying interface. It specifies a nondefault lockout time in the range 20 through 120 seconds.

```
[edit dynamic-profiles my-vlan-profile interfaces “$junos-interface-ifd-name” unit
“$junos-interface-unit” pppoe-underlying options]
user@host# set short-cycle-protection lockout-time-min 20 lockout-time-max 120 filter
aci
```

NOTE: If the ACI value is not present in the PPPoE attributes when you configure ACI-based subscriber session lockout, the router uses MAC-based lockout by default. With ACI-based encapsulation type lockout, PPPoE clients without an ACI attribute are also locked out.

RELATED DOCUMENTATION

PPPoE Subscriber Session Lockout Overview | 240
Clearing Lockout of PPPoE Subscriber Sessions | 250
Configuring an Underlying Interface for Dynamic PPPoE Subscriber Interfaces | 198
Configuring the PPPoE Family for an Underlying Interface | 201
Dynamic PPPoE Subscriber Interfaces over Static Underlying Interfaces Overview | 191

Clearing Lockout of PPPoE Subscriber Sessions

IN THIS SECTION

Purpose

Clear the lockout condition for the PPPoE subscriber session associated with a unique MAC source address or ACI value.

Action

- To clear the lockout condition for PPPoE subscriber sessions associated with all MAC source addresses on all underlying interfaces:

```
user@host> clear pppoe lockout
```

- To clear the lockout condition for the PPPoE subscriber session associated with the specified MAC source address:

```
user@host> clear pppoe lockout mac-address mac-address
```

- To clear the lockout condition for all PPPoE subscriber sessions on the specified underlying interface:

```
user@host> clear pppoe lockout underlying-interfaces underlying-interface-name
```

- To clear the lockout condition for the PPPoE subscriber session associated with the specified MAC source address on the specified underlying interface:

```
user@host> clear pppoe lockout mac-address mac-address underlying-interfaces underlying-
interface-name
```

- To clear the ACI-based lockout condition for PPPoE subscriber sessions on all underlying interfaces:

```
user@host> clear pppoe lockout aci
```

- To clear the ACI-based lockout condition for PPPoE subscriber sessions associated with the specified ACI value on the specified underlying interface:

```
user@host> clear pppoe lockout underlying-interfaces underlying-interface-name aci agent-
circuit-id
```

- To clear the ACI-based lockout for a PPPoE subscriber session with the specified ATM encapsulation type identifiers where the ACI value matches a regular expression:

```
user@host> clear pppoe lockout atm-identifier device-name device-name vpi vpi-identifier vci
vci-identifier aci "Relay-identifier atm 1/0:100\.*"
```

- To clear the MAC-based lockout condition for a PPPoE subscriber session with the specified ATM encapsulation type identifiers:

```
user@host> clear pppoe lockout atm-identifier device-name device-name vpi vpi-identifier vci
vci-identifier mac-address mac-address
```

- To clear the ACI-based lockout for a PPPoE subscriber session with the specified VLAN encapsulation type identifiers where the ACI value matches a regular expression:

```
user@host> clear pppoe lockout vlan-identifier device-name device-name svlan-id svlan-
identifier vlan-id vlan-identifier aci "Relay-identifier atm 3/0:200\.*"
```

- To clear the MAC-based lockout condition for a PPPoE subscriber session with the specified VLAN encapsulation type identifiers:

```
user@host> clear pppoe lockout vlan-identifier device-name device-name vlan-id vlan-
identifier mac-address mac-address
```

- To verify that the lockout condition has been cleared:

```
user@host> show pppoe lockout
```

RELATED DOCUMENTATION

CHAPTER 20

Configuring MTU and MRU for PPP Subscribers

IN THIS CHAPTER

Understanding MTU and MRU Configuration for PPP Subscribers

IN THIS SECTION

The default PPP maximum receive unit (MRU) and hence the maximum transmission unit (MTU) is 1500 bytes. For a PPPoE interface, the PPPoE header uses 6 bytes and the PPP protocol ID uses 2 bytes. This restricts the MRU size on a PPPoE interface to 1492 bytes, which can cause frequent fragmentation and reassembly of larger PPP packets received over the PPPoE interface. To override the default values, you can configure the MTU and MRU sizes for PPP subscribers.

For PPPoE subscribers, the PPP MRU or PPP MTU size can be greater than 1492 bytes if the PPP-Max-Payload tag is received in the PPPoE Active Discovery Request (PADR) packets or if the peer MRU received in the PPP LDP Configure-Request is greater than 1492 bytes. By default, PPPoE MTU and MRU are used on LNS if MTU or MRU or both are not explicitly configured.

The configuration of MRU and MTU is supported for subscribers of the following PPP connections:

- PPP over Ethernet (PPPoE) subscribers
- PPP over Ethernet over ATM (PPPoE over ATM) subscribers
- PPP over ATM (PPPoA) subscribers

- Tunneled PPP LAC subscribers
- Tunneled PPP LNS subscribers

PPP essentially negotiates between two independent half-duplex links. While establishing a PPP connection, PPP end-points negotiate the MRU to determine the PPP payload MTU on a negotiated PPP connection. The terms used in this section are described here:

Peer MRU MRU proposed by the peer to indicate the PPP payload size that it can accept.

PPP MRU MRU proposed by the router to indicate the PPP payload size that it can accept

PPP MTU PPP payload MTU (IP header + data) excluding any Layer 2 overhead.

By default, if the PPP MTU value is lower than 1492 bytes, the operational PPP MRU value is also set to the PPP MTU value. However, if the PPP MTU value is greater than 1492 bytes, Junos OS calculates the PPP MRU value based on the presence and value of the PPP-Max-Payload tag received in the PPPoE Active Discovery Request (PADR) packet. This default behavior can be changed by configuring the `mtu (`*`size`*`| use-lower-layer)` and `mru` *`size`* statements at the following hierarchy levels:

```
[edit access group-profile group-profile-name ppp ppp-options]
[edit dynamic-profiles profile-name interfaces pp0 unit “$junos-interface-unit” ppp-options],
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name"  unit “$junos-
interface-unit” ppp-options],
[edit interfaces pp0 unit unit-number ppp-options]
[edit interfaces si interface-id unit unit-number ppp-options]
```

PPP MTU and MRU for PPPoE Subscribers

For a PPPoE packet:

- Configured MTU is the MTU value configured using the `mtu` *`size`* statement.
- PPP lower-layer MTU is calculated as:interface MTU – [(Ethernet header payload) – (single-tagged VLANs) – (double-tagged VLANs) – (PPPoE header payload) – (PPP header)]

Junos OS determines the PPP MTU value for a terminated PPPoE interface based on the configured MTU, PPP lower-layer MTU, and the presence and value of the PPP-Max-Payload tag in the PADR packet and the peer's MRU.

1. If the PPP lower-layer MTU falls below 1492 bytes, then the PPP MTU value is the lesser of the PPP lower-layer MTU and the configured MTU value. The PPP-Max-Payload tag is ignored even if it is present in the PADR packet.

2. If the PPP lower-layer MTU is greater than 1492 bytes:

   - If the PPP-Max-Payload tag is not present in the PADR packet, then the PPP MTU value is the lesser of the configured MTU and the PPP lower-layer MTU value and the peer's MRU.

   - If the PPP-Max-Payload tag is present and its value is less than 1492 bytes, then the PPP MTU is the lesser of the configured MTU and the PPP lower-layer MTU value. Junos OS does not send out the PPP-Max-Payload tag in the PPPoE Active Discovery Session (PADS) packet to indicate that the router is not capable of supporting an MRU size greater than 1492 bytes.

   - If the PPP-Max-Payload tag is present and its value is greater than 1492 bytes but less than the configured MTU, the PPP MTU is the value received in the PPP-Max-Payload tag.

   - If the PPP-Max-Payload tag is present and its value is greater than 1492 bytes and also greater than the configured MTU, the PPP MTU is the lesser of the configured MTU, PPP lower-layer MTU value and the peer's MRU. Junos OS also returns the PPP-Max-Payload tag in the PADS packet to indicate that the router is capable of supporting an MRU greater than 1492 bytes.

By default, a router uses the PPP MTU value for the PPP MRU value during link control protocol (LCP) negotiation on point-to-point connections. When you configure the MRU for a PPP subscriber for PPPoE by using the `mru` *`size`* statement, Junos OS determines the PPP MRU value based on the following:

- If the MRU is configured using the `ppp-options` option, the PPP MRU is the lesser of the configured MRU value and the PPP MTU value for that subscriber (PPP MTU value derived based on the configured MTU, PPP lower-layer MTU, and the PPP-Max-Payload value in the PADR packet).

- If the MRU is not configured, the PPP MRU remains the same as the PPP MTU and is sent during LCP negotiation. During LCP negotiation, the server receives the peer MRU value and offers the PPP MRU derived from the configuration and the PPP MTU.

- For a negotiated PPP connection, PPP payload MTU (IP header + data) excluding any Layer 2 overhead, is set to the lesser of the PPP MTU and the received Peer MRU value.

PPP MTU and MRU for Tunneled Subscribers on LNS

For PPP subscribers on L2TP network server (LNS) you can explicitly configure the MTU and MRU on the inline service (si) interface to override the default values. When configured explicitly, the effective MTU can be either the explicit MTU size specified using the `mtu` *`size`* statement or the derived MTU using the `mtu use-lower-layer` statement.

- If the PPP MTU on the inline service (si) interface is configured as `use-lower-layer`, the PPP MTU is determined as:interface MTU – 58 bytes.

  58 bytes is the PPP overhead payload, which is calculated as the sum of the IP, UDP, L2TP, HDLC, and PPP header payloads.

- If the PPP MTU on the inline service (si) interface is configured using the `mtu` *`size`* statement, the PPP MTU is the lesser of the configured MTU, the (interface MTU – 58 bytes) value and the peer MRU learned via L2TP session negotiation or when renegotiating LCP with the remote peer.

When you configure an explicit MRU value by using the `mru` *`size`* statement, Junos OS determines the PPP MRU value for PPP subscribers on LNS interfaces based on the following scenarios:

- If the MRU value is not configured for PPP subscribers on the LNS and if the proxy LCP options are received from the L2TP access concentrator (LAC), the PPP MRU value offered in the LCP negotiation is the lesser of the PPP MTU and the proxy MRU value. If the LCP options are not received, PPP MTU is offered as MRU during LCP negotiation.
- If, however, the MRU value is configured for the PPP subscribers on the LNS, the PPP MRU is the lesser of the configured MRU and the PPP MTU value. Further, if the proxy LCP options are received from the LAC, the PPP MRU value sent during LCP negotiation is the lesser of the configured MRU or PPP MTU and the proxy MRU value.
- For a negotiated PPP payload MTU (IP header + data) excluding any Layer 2 overhead, the PPP MTU is set to the lesser of the PPP MTU and the received peer MRU value.

RELATED DOCUMENTATION

Configuring MTU and MRU for PPP Subscribers | 257

Configuring MTU and MRU for PPP Subscribers

You can configure the maximum transmission unit (MTU) and maximum receive unit for Point-to-Point Protocol (PPP) subscribers. This configuration is supported for the following PPP subscribers:

- PPP over Ethernet (PPPoE) subscribers
- PPP over Ethernet over ATM (PPPoE over ATM) subscribers
- PPP over ATM (PPPoA) subscribers
- Tunneled PPP LAC subscribers
- Tunneled PPP LNS subscribers

The MTU configuration specifies the maximum allowable data unit size (in bytes) that can be transmitted over a PPP connection without fragmentation. This size excludes the lower-layer header size. With this configuration, you can choose to either configure an explicit MTU value or use the MTU value configured for the interface excluding the lower-layer header size.

The MRU configuration specifies the size of maximum receive unit (MRU) that the router uses during link control protocol (LCP) negotiation for dynamic and static PPP subscribers and L2TP tunneled subscribers.

To configure MTU and MRU values for PPP subscribers:

- (Optional) Configure the MTU and the MRU for dynamic PPP subscribers (includes dynamic PPPoE and PPPoE over ATM subscribers).

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name"  unit “$junos-
interface-unit” ppp-options],
mru size;
user@host# mtu (size | use-lower-layer);
```

- (Optional) Configure the MTU and the MRU for static PPP subscribers (includes PPP over ATM subscribers).

```
[edit interfaces pp0 unit unit-number ppp-options]
mru size;
user@host# mtu (size | use-lower-layer);
```

- (Optional) Configure the MTU and the MRU for dynamic tunneled PPP subscribers for L2TP LNS.

```
[edit dynamic-profiles profile-name interfaces "$junos-interface-ifd-name"  unit “$junos-
interface-unit” ppp-options],
mru size;
user@host# mtu (size | use-lower-layer);
```

- (Optional) Configure the MTU and the MRU for static tunneled PPP subscribers for L2TP LNS.

```
[edit interfaces si interface-id unit unit-number ppp-options]
mru size;
user@host# mtu (size | use-lower-layer);
```

- Configure the MTU and the MRU for static and dynamic PPP subscribers associated with a group profile.

```
[edit access group-profile group-profile-name ppp ppp-options]
mru size;
user@host# mtu (size | use-lower-layer);
```

RELATED DOCUMENTATION

Understanding MTU and MRU Configuration for PPP Subscribers | 254

CHAPTER 21

Configuring PPPoE Service Name Tables

IN THIS CHAPTER

Understanding PPPoE Service Name Tables

IN THIS SECTION

On an MX Series router acting as a remote access concentrator (AC), also referred to as a *PPPoE server*, you can configure up to 32 PPPoE service name tables and assign the service name tables to PPPoE underlying interfaces. A *PPPoE service name table* defines the set of *services* that the router can provide to a PPPoE client. Service entries configured in a PPPoE service name table represent the *service name tags* transmitted between the client and the router in a PPPoE control packet.

This overview covers the following topics to help you understand and configure PPPoE service name tables:

Interaction Among PPPoE Clients and Routers During the Discovery Stage

In networks with mesh topologies, PPPoE clients are often connected to multiple PPPoE servers (remote ACs). During the PPPoE discovery stage, a PPPoE client identifies the Ethernet MAC address of the remote AC that can service its request, and establishes a unique PPPoE session identifier for a connection to that AC.

The following steps describe, at a high level, how the PPPoE client and the remote AC (router) use the PPPoE service name table to interact during the PPPoE discovery stage:

1. The PPPoE client broadcasts a PPPoE Active Discovery Initiation (PADI) control packet to all remote ACs in the network to request that an AC support certain services.

   The PADI packet must contain either, but not both, of the following:

   - One and only one nonzero-length service name tag that represents a specific client service
   - One and only one empty (zero-length) service name tag that represents an unspecified service

2. One or more remote ACs respond to the PADI packet by sending a PPPoE Active Discovery Offer (PADO) packet to the client, indicating that the AC can service the client request.

   To determine whether it can service a particular client request, the router matches the service name tag received in the PADI packet against the service name tags configured in its service name table. If a matching service name tag is found in the PPPoE service name table, the router sends the client a PADO packet that includes the name of the AC from which it was sent. If no matching service name tag is found in the PPPoE service name table, the router drops the PADI request and does not send a PADO response to the client.

3. The PPPoE client sends a unicast PPPoE Active Discovery Request (PADR) packet to the AC to which it wants to connect, based on the responses received in the PADO packets.

4. The selected AC sends a PPPoE Active Discovery Session (PADS) packet to establish the PPPoE connection with the client.

Service Entries and Actions in PPPoE Service Name Tables

A PPPoE service name table can include three types of service entries: named services, an `empty` service, and an `any` service. For each service entry, you specify the action to be taken by the underlying interface when the router receives a PADI packet containing the specified service name tag.

You can configure the following services and actions in a PPPoE service name table:

- Named service—Specifies a PPPoE client service that an AC can support. For example, you might configure named services associated with different subscribers who log in to the PPPoE server, such as `user1-service` or `user2-service`, or that correspond to different ISP service level agreements, such as `premium` and `standard`. Each PPPoE service name table can include a maximum of 512 named service entries, excluding `empty` and `any` service entries. A named service is associated with the `terminate` action by default.

- `empty` service—A service tag of zero length that represents an unspecified service. Each PPPoE service name table includes one empty service. The `empty` service is associated with the `terminate` action by default.

- `any` service—Acts as a default service for non-empty service entries that do not match the named service entries or `empty` service entry configured in the PPPoE service name table. Each PPPoE service name table includes one `any` service. The `any` service is useful when you want to match the agent circuit identifier and agent remote identifier information for a PPPoE client, but do not care about the contents of the service name tag transmitted in the control packet. The `any` service is associated with the `drop` action by default.

- Action—Specifies the action taken by the underlying PPPoE interface assigned to the PPPoE service name table on receipt of a PADI packet from the client containing a particular service request. You can configure one of the following actions for the associated named service, `empty` service, `any` service, or agent circuit identifier/agent remote identifier (ACI/ARI) pair in the PPPoE service name table on the router:

  - `terminate`—(Default) Directs the router to immediately respond to the PADI packet by sending the client a PADO packet containing the name of the AC that can service the request. Named services, `empty` services, and ACI/ARI pairs are associated with the `terminate` action by default. Configuring the `terminate` action for a service enables you to more tightly control which PPPoE clients can access and receive services from a particular PPPoE server.

- `delay`—Number of seconds that the PPPoE underlying interface waits after receiving a PADI packet from the client before sending a PADO packet in response. In networks with mesh topologies, you might want to designate a primary PPPoE server and a backup PPPoE server for handling a particular service request. In such a scenario, you can configure a delay for the associated service entry on the backup PPPoE server to allow sufficient time for the primary PPPoE server to respond to the client with a PADO packet. If the primary server does not send the PADO packet within the delay period configured on the backup server, then the backup server sends the PADO packet after the delay period expires.

- `drop`—Directs the router to drop (ignore) a PADI packet containing the specified service name tag when received from a PPPoE client, which effectively denies the client's request to provide the associated service. The `any` service is associated with the `drop` action by default. To prohibit the router from responding to PADI packets that contain `empty` or `any` service name tags, you can configure the `drop` action for the empty or `any` service. You can also use the `drop` action in combination with ACI/ARI pairs to accept specific service name tags only from specific subscribers, as described in the following information about ACI/ARI pairs.

ACI/ARI Pairs in PPPoE Service Name Tables

To specify agent circuit identifier (ACI) and agent remote identifier (ARI) information for a named service, `empty` service, or `any` service in a PPPoE service name table, you can configure an ACI/ARI pair. An ACI/ARI pair contains an agent circuit ID string that identifies the DSLAM interface that initiated the service request, and an agent remote ID string that identifies the subscriber on the DSLAM interface that initiated the service request. You can think of an ACI/ARI pair as the representation of one or more PPPoE clients accessing the router by means of the PPPoE service name table.

ACI/ARI specifications support the use of wildcard characters in certain formats. You can configure a combined maximum of 8000 ACI/ARI pairs, both with and without wildcards, per PPPoE service name table. You can distribute the ACI/ARI pairs in any combination among the service entries in the service name table.

You must specify the action—`terminate`, `delay`, or `drop`—taken by the underlying PPPoE interface when it receives a client request containing vendor-specific ACI/ARI information that matches the ACI/ARI information configured in the PPPoE service name table on the router. An ACI/ARI pair is associated with the `terminate` action by default.

For example, assume that for the `user1-service` named service, you configure the `drop` action for the service and the `terminate` action for the associated ACI/ARI pairs. In this case, the ACI/ARI pairs identify the DSLAM interfaces and associated subscribers authorized to access the PPPoE server. Using this configuration causes the router to drop PADI packets containing the `user1-service` tag *unless* the PADI packet also contains vendor-specific ACI/ARI information that matches the subscribers identified in one or more of the ACI/ARI pairs. For PADI packets containing matching ACI/ARI information, the router sends an immediate PADO response to the client indicating that it can provide the requested service for the specified subscribers.

You can also associate a PPPoE dynamic profile, routing instance, and static PPPoE interface with an ACI/ARI pair.

Dynamic Profiles and Routing Instances in PPPoE Service Name Tables

You can associate a previously configured PPPoE dynamic profile with a named service, `empty` service, or `any` service in the PPPoE service name table, or with an ACI/ARI pair defined for these services. The router uses the attributes defined in the profile to instantiate a dynamic PPPoE subscriber interface based on the service name, ACI, and ARI information provided by the PPPoE client during PPPoE negotiation. The dynamic profile configured for a service entry or ACI/ARI pair in a PPPoE service name table overrides the dynamic profile assigned to the PPPoE underlying interface on which the dynamic PPPoE interface is created.

To specify the routing instance in which to instantiate the dynamic PPPoE interface, you can associate a previously configured routing instance with a named service, `empty` service, or `any` service in the PPPoE service name table, or with an ACI/ARI pair defined for these services. Like dynamic profiles configured for service entries or ACI/ARI pairs, the routing instance configured for the PPPoE service name table overrides the routing instance assigned to the PPPoE underlying interface.

For information about configuring the PPPoE service name table to create a dynamic PPPoE subscriber interface, see "Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation" on page 275.

Maximum Sessions Limit in PPPoE Service Name Tables

To limit the number of PPPoE client sessions that can use a particular service entry in the PPPoE service name table, you can configure the maximum number of active PPPoE sessions using either dynamically-created or statically-created PPPoE interfaces that the router can establish with a particular named service, `empty` service, or `any` service. (You cannot configure the maximum sessions limit for an ACI/ARI pair.) The maximum sessions limit must be in the range 1 through the platform-specific maximum PPPoE sessions supported for your routing platform. The router maintains a count of active PPPoE sessions for each service entry to determine when the maximum sessions limit has been reached.

The router uses the maximum sessions value for a service entry in the PPPoE service name table in conjunction with both of the following:

- The maximum sessions (`max-sessions`) value configured for the PPPoE underlying interface
- The maximum number of PPPoE sessions supported on your routing platform

If your configuration exceeds either of these maximum session limits, the router cannot establish the PPPoE session.

Static PPPoE Interfaces in PPPoE Service Name Tables

To reserve a previously configured static PPPoE interface for use only by the PPPoE client with matching ACI/ARI information, you can specify a single static PPPoE interface for each ACI/ARI pair defined for a named service entry, `empty` service entry, or `any` service entry in a PPPoE service name table. (You cannot configure a static interface for a service entry that does not have an ACI/ARI pair defined.) The static PPPoE interface associated with an ACI/ARI pair takes precedence over the general pool of static PPPoE interfaces associated with the PPPoE underlying interface configured on the router.

When you configure a static interface in the PPPoE service name table, make sure there is a one-to-one correspondence between the PPPoE client and the static interface. For example, if two clients have identical ACI/ARI information that matches the information in the PPPoE service name table, the router reserves the static interface for exclusive use by the first client that logs in to the router. As a result, the router prevents the second client from logging in.

You cannot configure a static interface for an ACI/ARI pair already configured with a dynamic profile and routing instance. Conversely, you cannot configure a dynamic profile and routing instance for an ACI/ARI pair already configured with a static interface.

PADO Advertisement of Named Services in PPPoE Service Name Tables

By default, the advertisement of named services in PADO control packets sent by the router to the PPPoE client is disabled. You can enable advertisement of named services in the PADO packet as a global option when you configure the PPPoE protocol on the router. Configuring PADO advertisement notifies PPPoE clients of the services that the router (AC) can offer.

If you enable advertisement of named services in PADO packets, make sure the number and length of all advertised service entries does not exceed the maximum transmission unit (MTU) size supported by the PPPoE underlying interface.

Limiting the subscriber sessions per AE or PFE Bundle in PPPoE Service Name Tables

The PPPoE Service-Name table functionality may be used to limit the number of PPPoE subscriber sessions per PFE or AE bundle. This is accomplished by configuring all PPPoE underlying VLAN interfaces over a specific PFE or AE bundle with a single Service-Name table. This Service-Name table should contain only the service “any” with a max-sessions value equal to the PPPoE subscriber session limit for the PFE or AE bundle. The each PFE or AE bundle must have its own unique Service-Name table to ensure that PPPoE subscribers from other PFE or AE bundles are not incorrectly counted against a PFE or AE-specific session limit.

To configure a service-name table for PPPoE sessions on underlying VLAN interfaces to limit the number of subscriber sessions per PFE or AFE bundle, include the set service-name-table <PFE/AE-table-name> service any max-sessions <PPPoE-subscriber-limit> statement at the `[edit protocols pppoe]` hierarchy level.

RELATED DOCUMENTATION

Evaluation Order for Matching Client Information in PPPoE Service Name Tables | 266

Benefits of Configuring PPPoE Service Name Tables | 267

Configuring PPPoE Service Name Tables | 269

Example: Configuring a PPPoE Service Name Table | 278

Configuring Dynamic PPPoE Subscriber Interfaces | 194

PPPoE Overview

Evaluation Order for Matching Client Information in PPPoE Service Name Tables

When the router receives a service request from a PPPoE client, it evaluates the entries configured in the PPPoE service name table to find a match for the client's ACI/ARI information so it can take the appropriate action.

The order of evaluation is as follows:

1. The router evaluates the ACI/ARI information configured for the `any` service entry, and ignores the contents of the service name tag transmitted by the client.

2. If no match is found for the client information, the router evaluates the ACI/ARI information for the `empty` service entry and the named service entries. If an ACI/ARI pair is not configured for these service entries, the router evaluates the other attributes configured for the `empty` service and named services.

3. If there is still no match for the client information, the router evaluates the other attributes configured for the `any` service entry, and ignores both the ACI/ARI information for the `any` service and the contents of the service name tag transmitted by the client. If the `any` service is configured for the default action, `drop`, the router drops the PADR packet. If the `any` service is configured for a nondefault action (`terminate` or `delay`), the router evaluates the other attributes configured for the `any` service.

RELATED DOCUMENTATION

Benefits of Configuring PPPoE Service Name Tables

This topic describes the benefits of configuring PPPoE service name tables.

Configuring PPPoE service name tables provides the following benefits:

- Enables support for multiple services requested by PPPoE clients, and configuration of an action for the underlying PPPoE interface to take (`delay`, `drop`, or `terminate`) upon receipt of a PPPoE Active Discovery Initiation (PADI) packet requesting that service.
- Provides tighter control over which PPPoE clients can log in to and receive services from a particular PPPoE server.
- Provides load balancing across a set of remote access concentrators (ACs) in a mesh topology by enabling you to configure agent circuit identifier/agent remote identifier (ACI/ARI) pairs for named, `empty`, and `any` service entries to specify the appropriate AC to receive and service a particular PPPoE client request.
- Offers a more targeted approach to configuration of PPPoE sessions based on the service name and ACI/ARI information provided by the PPPoE client during PPPoE negotiation.
- Supports creation of dynamic PPPoE subscriber interfaces in a specified routing instance based on configuration of a service entry or ACI/ARI pair in the PPPoE service name table.
- Enables you to reserve a specified static PPPoE interface for use only by the PPPoE client with matching ACI/ARI information.
- Enables you to specify the maximum number of PPPoE client sessions that can use a particular service entry in the PPPoE service name table.
- Provides redundancy across a set of remote ACs in a mesh topology by enabling you to configure a primary AC and a backup AC for handling a specific service request from a PPPoE client.

For example, on the primary AC for handling a client service, you might configure the `terminate` action for the associated service to direct the primary AC to immediately send a PPPoE Active Discovery Offer (PADO) packet in response to a PADI packet containing that service name tag. On the backup AC for the client service, you might configure the `delay` action for the associated service to specify the number of seconds the backup AC waits after receiving a PADI packet from the client before sending a PADO packet in response. If the primary AC does not send a PADO packet to the client within the delay period configured on the backup AC, then the backup AC sends the PADO packet after the delay period expires.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260
Configuring PPPoE Service Name Tables | 269
Example: Configuring a PPPoE Service Name Table | 278
PPPoE Overview
Ethernet Interfaces User Guide for Routing Devices

Creating a Service Name Table

You can create up to 32 PPPoE service name tables on the router. You can optionally create named services and add them to a service name table. By default, the `empty` service and the `any` service are present in each service name table.

A named service specifies a PPPoE client service that the router, functioning as an access concentrator or PPPoE server, can support. The `empty` service is a service tag of zero length that represents an unspecified service. The `any` service acts as a default service for non-empty service entries that do not match the named or `empty` service entries configured in the PPPoE service name table. Named services and the `empty` service are associated with the `terminate` action by default, and the `any` service is associated with the `drop` action by default.

To create a PPPoE service name table:

- Specify the table name.

```
[edit protocols pppoe]
user@host# set service-name-tables table1
```

RELATED DOCUMENTATION

Configuring PPPoE Service Name Tables | 269
Understanding PPPoE Service Name Tables | 260
PPPoE Overview

Configuring PPPoE Service Name Tables

To configure PPPoE service name tables:

1. Create a PPPoE service name table.

   See "Creating a Service Name Table" on page 268.

2. (Optional) Configure the action taken for the `empty` service.

   See "Configuring the Action Taken When the Client Request Includes an Empty Service Name Tag" on page 271.

3. (Optional) Configure the action taken for the `any` service.

   See "Configuring the Action Taken for the Any Service" on page 272.

4. Assign a named service to the service name table and optionally configure the action taken for the specified service name.

   See "Assigning a Service to a Service Name Table and Configuring the Action Taken When the Client Request Includes a Non-zero Service Name Tag" on page 273.

5. (Optional) Configure the action taken for an ACI/ARI pair associated with a service.

   See "Assigning an ACI/ARI Pair to a Service Name and Configuring the Action Taken When the Client Request Includes ACI/ARI Information" on page 274.

6. (Optional) Assign a dynamic profile and routing instance to a service name or ACI/ARI pair to instantiate a dynamic PPPoE subscriber interface.

   See "Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation" on page 275.

7. (Optional) Limit the number of active PPPoE sessions that the router can establish with the specified service.

   See "Limiting the Number of Active PPPoE Sessions Established with a Specified Service Name" on page 277.

8. (Optional) Assign a static PPPoE interface to an ACI/ARI pair to reserve the interface for exclusive use by the PPPoE client with matching ACI/ARI information.

   See "Reserving a Static PPPoE Interface for Exclusive Use by a PPPoE Client" on page 277.

9. (Optional) Enable advertisement of named services in the PADO control packet sent by the router to the client.

   See "Enabling Advertisement of Named Services in PADO Control Packets" on page 289.

10. Assign a service name table to a PPPoE underlying interface.

    See "Assigning a Service Name Table to a PPPoE Underlying Interface" on page 270.

11. (Optional) Configure trace options for troubleshooting the configuration.

    See Tracing PPPoE Operations.

RELATED DOCUMENTATION

Assigning a Service Name Table to a PPPoE Underlying Interface

You must assign the PPPoE service name table to a PPPoE underlying interface.

Before you begin:

- Specify PPPoE as the encapsulation method on the underlying interface.

  See *Setting the Appropriate Encapsulation on the PPPoE Interface* in Configuring PPPoE.

To assign a service name table to a PPPoE underlying interface:

- Specify the table name:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set pppoe-underlying-options service-name-table table1
```

RELATED DOCUMENTATION

Configuring the Action Taken When the Client Request Includes an Empty Service Name Tag

You can configure the action taken by the PPPoE underlying interface when it receives a PADI packet that includes a zero-length (empty) service name tag. The `empty` service is present by default in every PPPoE service name table.

To indicate that it can service the client request, the interface returns a PADO packet in response to the PADI packet. By default, the interface immediately responds to the request; this is the `terminate` action. Alternatively, you can configure the `drop` action to ignore (drop) the PADI packet, or the `delay` action to set a delay between receipt of the PADI packet and transmission of the PADO packet.

(Optional) To configure the action taken for the `empty` service in response to a PADI packet from a PPPoE client:

- Specify the action.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service empty drop
```

You can also accomplish the following optional tasks when you configure the `empty` service:

- Specify the agent circuit identifier (ACI) and agent remote identifier (ARI) information to determines the action taken by the PPPoE underlying interface when it receives a PADI packet with matching ACI/ARI information.
- Specify a dynamic profile and routing instance with which the router instantiates a dynamic PPPoE subscriber interface.
- Limit the number of active PPPoE sessions that the router can establish with the `empty` service.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260

Configuring PPPoE Service Name Tables | 269

Assigning an ACI/ARI Pair to a Service Name and Configuring the Action Taken When the Client Request Includes ACI/ARI Information | 274

Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation | 275

Limiting the Number of Active PPPoE Sessions Established with a Specified Service Name | 277

PPPoE Overview

Configuring the Action Taken for the Any Service

The any service acts as a default service for service name tags transmitted by the client that do not match any of the service entries configured in the PPPoE service name table on the router. By configuring an action for the any service, you specify the action taken by the PPPoE underlying interface when it receives a PADI control packet from a client that includes a non-empty service name tag that does not match any of the named service entries or empty service entry in the PPPoE service name table.

Each PPPoE service name table includes one any service entry associated by default with the drop action. The drop action ignores a PADI packet containing a nonmatching service name tag. Alternatively, you can configure the terminate action to immediately respond to the PADI packet with a PADO packet, or the delay action to specify a delay between receipt of the PADI packet and transmission of the PADO packet.

To configure the action taken for the any service in response to a PADI packet from a PPPoE client:

- Specify the action.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service any terminate
```

You can also accomplish the following optional tasks when you configure the any service:

- Specify the agent circuit identifier (ACI) and agent remote identifier (ARI) information to determine the action taken by the PPPoE underlying interface when it receives a PADI packet with matching ACI/ARI information.
- Specify a dynamic profile and routing instance with which the router instantiates a dynamic PPPoE subscriber interface.
- Limit the number of active PPPoE sessions that the router can establish with the any service.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260

Configuring PPPoE Service Name Tables | 269

Assigning an ACI/ARI Pair to a Service Name and Configuring the Action Taken When the Client Request Includes ACI/ARI Information | 274

Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation | 275

Limiting the Number of Active PPPoE Sessions Established with a Specified Service Name | 277

PPPoE Overview

Assigning a Service to a Service Name Table and Configuring the Action Taken When the Client Request Includes a Non-zero Service Name Tag

You can configure a maximum of 512 named service entries, excluding `empty` and `any` service entries, across all PPPoE service name tables on the router. A named service specifies a PPPoE client service that the router, functioning as an access concentrator or PPPoE server, can support. You can optionally configure the action taken by the PPPoE underlying interface when it receives a PADI packet that includes a matching named service (service name tag).

To indicate that it can service the client request, the interface returns a PADO packet in response to the PADI packet. By default, the interface immediately responds to the request; this is the `terminate` action. Alternatively, you can configure the `drop` action to ignore (drop) the PADI packet, or the `delay` action to set a delay between receipt of the PADI packet and transmission of the PADO packet.

(Optional) To configure a named service for a PPPoE service name table, do one of the following:

- Assign a service name to the table. The `terminate` action is applied to the service by default.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service gold-service
```

- Specify the action taken for a service in response to a PADI packet from a PPPoE client.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service gold-service delay 25
```

You can also accomplish the following optional tasks when you configure a named service:

- Specify the agent circuit identifier (ACI) and agent remote identifier (ARI) information to determines the action taken by the PPPoE underlying interface when it receives a PADI packet with matching ACI/ARI information.
- Specify a dynamic profile and routing instance with which the router instantiates a dynamic PPPoE subscriber interface.
- Limit the number of active PPPoE sessions that the router can establish with the specified named service.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260

Assigning an ACI/ARI Pair to a Service Name and Configuring the Action Taken When the Client Request Includes ACI/ARI Information

You can configure up to 8000 agent circuit identifier/agent remote identifier (ACI/ARI) pairs per PPPoE service name table, distributed in any combination among the named, `empty`, and `any` service entries in the service name table. You can optionally configure the action taken by the PPPoE underlying interface when it receives a PADI packet that includes a service name tag and the vendor-specific tag with ACI/ARI information that matches the ACI/ARI pair that you specify.

You can use an asterisk (*) as a wildcard character to match ACI/ARI pairs, the ACI alone, or the ARI alone. The asterisk can be placed only at the beginning, the end, or both the beginning and end of the identifier string. You can also specify an asterisk alone for either the ACI or the ARI. You cannot specify only an asterisk for both the ACI and the ARI. When you specify a single asterisk as the identifier, that identifier is ignored in the PADI packet.

For example, suppose you care about matching only the ACI and do not care what value the ARI has in the PADI packet, or even whether the packet contains an ARI value. In this case you can set the *remote-id-string* to a single asterisk. Then the interface ignores the ARI received in the packet and the interface takes action based only on matching the specified ACI.

To indicate that it can service the client request, the interface returns a PADO packet in response to the PADI packet. By default, the interface immediately responds to the request; this is the `terminate` action. Alternatively, you can configure the `drop` action to ignore (drop) the PADI packet, or the `delay` action to set a delay between receipt of the PADI packet and transmission of the PADO packet.

To configure an ACI/ARI pair for a named, `empty`, or `any` service, do one of the following:

- Assign an ACI/ARI pair to the service name. The `terminate` action is applied to the pair by default.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service gold-service agent-specifier aci DSLAM:3/0/1/101 ari *user*
```

- Specify the action taken for the ACI/ARI pair in response to a PADI packet from a PPPoE client.

```
[edit protocols pppoe service-name-tables table1]
user@host# set service any agent-specifier aci velorum-ge-2/0/3 ari westford delay 90
```

In this example, an ACI/ARI pair and the `delay` action are configured for the `any` service. Configuring an ACI/ARI pair for the `any` service is useful when you want to match the agent circuit identifier and agent remote identifier information for a specific PPPoE client, but do not care about the contents of the service name tag transmitted by the client in the PADI packet.

You can also accomplish the following optional tasks when you configure an ACI/ARI pair:

- Specify a dynamic profile and routing instance with which the router instantiates a dynamic PPPoE subscriber interface.
- Reserve a specified static PPPoE interface for exclusive use by the PPPoE client with match ACI/ARI information.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260
Configuring PPPoE Service Name Tables | 269
Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation | 275
Reserving a Static PPPoE Interface for Exclusive Use by a PPPoE Client | 277
PPPoE Overview

Assigning a Dynamic Profile and Routing Instance to a Service Name or ACI/ARI Pair for Dynamic PPPoE Interface Creation

You can create a dynamic PPPoE subscriber interface based on the service name, agent circuit identifier (ACI), and agent remote identifier (ARI) information provided by the PPPoE client during PPPoE negotiation. To do so, you assign a PPPoE dynamic profile to a named service, `empty` service, or `any` service entry in a PPPoE service name table, or to an ACI/ARI pair defined for these services.

Similarly, to specify the routing instance in which to instantiate the dynamic PPPoE subscriber interface, you can assign a routing instance to a named service, `empty` service, or `any` service in a PPPoE service name table, or to an ACI/ARI pair defined for these services.

Observe the following configuration guidelines when you assign a dynamic profile and routing instance to a PPPoE service name table to create a dynamic PPPoE subscriber interface:

- The dynamic profile or routing instance assigned to the PPPoE service name table overrides the dynamic profile or routing instance assigned to the PPPoE underlying interface on which the dynamic subscriber interface is created.
- You cannot configure a dynamic profile or routing instance for an ACI/ARI pair already configured with a static interface (by using the `static-interface` statement). Conversely, you cannot configure a static interface for an ACI/ARI pair already configured with a dynamic profile or routing instance.

Before you begin:

1. Configure a PPPoE dynamic profile.

   To configure a basic PPPoE dynamic profile, see "Configuring a PPPoE Dynamic Profile" on page 195.

2. Configure the routing instance in which you want the router to instantiate the dynamic profile.

   For information about configuring routing instances, see Routing Instances Overview.

3. Create the PPPoE service name table on the router.

   See "Creating a Service Name Table" on page 268.

To create a dynamic PPPoE subscriber interface based on the service name and, optionally, associated ACI/ARI pair configured in a PPPoE service name table, do one of the following:

- Assign a previously configured dynamic profile and routing instance to a named, `empty`, or `any` service.

  ```
  [edit protocols pppoe service-name-tables table1]
  user@host# set service premium dynamic-profile premiumProfile routing-instance premiumRI
  ```

- Assign a previously configured dynamic profile and routing instance to the ACI/ARI pair defined for a named, `empty`, or `any` service.

  ```
  [edit protocols pppoe service-name-tables table1]
  user@host# set service any agent-specifier aci west-ge-3/0/3 ari sunnyvale dynamic-profile
  standardProfile routing-instance standardRI
  ```

RELATED DOCUMENTATION

Example: Configuring a PPPoE Service Name Table for Dynamic Subscriber Interface Creation | 282

Limiting the Number of Active PPPoE Sessions Established with a Specified Service Name

To limit the number of PPPoE client sessions that can use a particular service entry in the PPPoE service name table, you can configure the maximum number of PPPoE sessions using static or dynamic PPPoE interfaces that the router can establish with the specified named service, `empty` service, or `any` service. You cannot configure a maximum sessions limit for an ACI/ARI pair in the service name table.

The maximum sessions limit must be in the range 1 through the platform-specific maximum PPPoE sessions supported for your routing platform. The router maintains a count of active PPPoE sessions for each service entry to determine when the maximum sessions limit has been reached.

To limit the number of PPPoE client sessions for a particular named, `empty`, or `any` service:

- Configure the maximum sessions limit for the specified service:

```
[edit protocols pppoe service-name-tables tableEast]
user@host# set service premium-service max-sessions 100
```

RELATED DOCUMENTATION

Reserving a Static PPPoE Interface for Exclusive Use by a PPPoE Client

To reserve a static PPPoE interface for exclusive use by the PPPoE client with matching agent circuit identifier/agent remote identifier (ACI/ARI) information, you can assign a static PPPoE interface to an ACI/ARI pair defined for a named service entry, `empty` service entry, or `any` service entry in a PPPoE service name table. You cannot assign a static PPPoE interface directly to a service entry that does not have an ACI/ARI pair defined.

Observe the following guidelines when you configure a static PPPoE interface for an ACI/ARI pair:

- You can specify only one static PPPoE interface per ACI/ARI pair.
- If the ACI/ARI pair represents an individual PPPoE client, make sure there is a one-to-one correspondence between the client and the static PPPoE interface.
- The static interface associated with the ACI/ARI pair takes precedence over the general pool of static interfaces associated with the PPPoE underlying interface.
- You cannot configure a static interface for an ACI/ARI pair already configured with a dynamic profile and routing instance. Conversely, you cannot configure a dynamic profile and routing instance for an ACI/ARI pair already configured with a static interface.

Before you begin:

- Configure the static PPPoE interface.

  See Configuring PPPoE.

To reserve a static PPPoE interface for exclusive use by the PPPoE client with matching ACI/ARI information:

- Assign a previously configured static PPPoE interface to the ACI/ARI pair defined for a named, `empty`, or `any` service entry:

```
[edit protocols pppoe service-name-tables tableEast]
user@host# set service any agent-specifier aci velorum-ge-2/0/3 ari westford static-interface
pp0.100
```

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260
Configuring PPPoE Service Name Tables | 269
PPPoE Overview

Example: Configuring a PPPoE Service Name Table

This example shows how you can configure a PPPoE service name table on an MX Series router with service entries that correspond to different client services. By configuring the appropriate actions (`delay`, `terminate`, or `drop`) and agent circuit identifier/agent remote identifier (ACI/ARI) pairs for the service entries, you can provide load balancing and redundancy across a set of remote access concentrators (ACs) in a mesh topology, and determine how best to allocate service requests from PPPoE clients to the servers in your network.

In this example, the PPPoE service name table, Table1, contains the following service entries:

- `user1-service`—Named service representing the subscriber service for user1.
- `user2-service`—Named service representing the subscriber service for user2.
- `empty` service—Represents an unspecified service.

To configure a PPPoE service name table with service entries that correspond to different subscriber services:

1. Create the PPPoE service name table and define the services and associated actions.

```
[edit protocols pppoe]
service-name-tables Table1 {
    service empty {
        drop;
    }
    service user1-service {
        terminate;
        agent-specifier {
            aci “east*” ari “wfd*” delay 10;
            aci “west*” ari “svl*” delay 10;
        }
    }
    service user2-service {
        delay 20;
    }
}
```

This example creates a PPPoE service name table named Table1 with three service entries, as follows:

- The `empty` service is configured with the `drop` action. This action prohibits the router (AC) from responding to PADI packets from the client that contain empty service name tags.
- The `user1-service` named service is configured with both the `terminate` action, and two ACI/ARI (agent-specifier) pairs:
    - The `terminate` action directs the router to immediately respond to PADI packets from the client that contain the `user1-service` tag, and is the default action for named services.

- The 10-second delay configured for each ACI/ARI pair applies only to PADI packets from the client that contains a vendor-specific tag with matching ACI and ARI information. In this example, configuring the `delay` action indicates that the `east` or `west` server is considered the backup AC for handling these client requests, and that you expect an AC other than `east` or `west` to handle the request as the primary server. If the primary AC does not respond to the client with a PADO packet within 10 seconds, then the `east` or `west` backup AC sends the PADO packet after the 10-second delay expires.

- The `user2-service` named service is configured with a 20-second delay, indicating that you expect an AC other than the one on which this PPPoE service name table is configured to be the primary AC for handling this client request. If the primary AC does not respond to the client with a PADO packet within 20 seconds, then the backup AC (that is, the router on which you are configuring the service name table) sends the PADO packet after the 20-second delay expires.

2. Assign the PPPoE service name table to a PPPoE underlying interface configured with PPPoE encapsulation.

```
[edit interfaces]
ge-2/0/3 {
    vlan-tagging;
    unit 0 {
        vlan-id 100;
        encapsulation ppp-over-ethernet;
        pppoe-underlying-options {
            service-name-table Table1;
        }
    }
}
```

3. (Optional) Verify the PPPoE service name table configuration.

```
user@host> show pppoe service-name-tables Table1
Service Name Table: Table1
  Service Name: <empty>
  Service Action: Drop

  Service Name: user1-service
  Service Action: Terminate
    ACI: east*
    ARI: wfd*
      ACI/ARI Action: Delay 10 seconds
    ACI: west*
```

```
    ARI: svl*
      ACI/ARI Action: Delay 10 seconds

  Service Name: user2-service
  Service Action: Delay 20 seconds
```

4. (Optional) Verify whether the PPPoE service name table has been properly assigned to the underlying PPPoE interface, and whether packet transfer between the router (AC) and PPPoE client is working correctly.

```
user@host>show pppoe underlying-interfaces ge-2/0/3.0 extensive
ge-2/0/3.0 Index 72
  State: Static, Dynamic Profile: None,
  Max Sessions: 4000, Active Sessions: 2,
  Service Name Table: Table1, Duplicate Protection: Off,
  AC Name: east
  PacketType                         Sent          Received
    PADI                                0                 2
    PADO                                2                 0
    PADR                                0                 2
    PADS                                2                 0
    PADT                                0                 1
    Service name error                  0                 0
    AC system error                     0                 0
    Generic error                       0                 0
    Malformed packets                   0                 0
    Unknown packets                     0                 0
```

Examine the command output to ensure the following:

- The `Service Name Table` field displays the name of the correct PPPoE service name table. This field displays `none` if no service name table has been associated with the specified interface.
- The `Sent` and `Received` values for the `Service name error` field are 0 (zero). For example, a nonzero value in the `Received` field for `Service name error` indicates that there are errors in the control packets received from PPPoE clients, such as a PADI packet that does not contain a service name tag.

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260
Configuring PPPoE Service Name Tables | 269

Troubleshooting PPPoE Service Name Tables | 286
PPPoE Overview

Example: Configuring a PPPoE Service Name Table for Dynamic Subscriber Interface Creation

This example shows how to configure a PPPoE service name table to create a dynamic PPPoE subscriber interface based on the service name, agent circuit identifier (ACI), and agent remote identifier (ARI) information provided by PPPoE clients during PPPoE negotiation.

In this example, PPPoE service name table `TableDynamicPPPoE` includes an `any` service entry, `empty` service entry, and two named service entries: `Premium` and `Standard`. The PPPoE underlying interfaces configured for `TableDynamicPPPoE` are `ge-2/0/0.1` and `ge-2/0/0.2`. Only `ge-2/0/0.1` is configured for dynamic profile assignment and creation of dynamic PPPoE subscriber interfaces.

Following the configuration example, Table 11 on page 284 explains how the router evaluates the entries in `TableDynamicPPPoE` to create a dynamic PPPoE subscriber interface in a specified routing instance for each of several sample clients.

To configure a PPPoE service name table to create dynamic PPPoE subscriber interfaces:

1. Configure the PPPoE service name table.

```
protocols {
    pppoe {
        service-name-tables TableDynamicPPPoE {
            service any {
                terminate;
                max-sessions 100;
                dynamic-profile AnyProfile;
                agent-specifier {
                    aci "broadway-ge-1/0/1.0" ari "london" {
                        terminate;
                        dynamic-profile LondonProfile;
                        routing-instance LondonRI;
                    }
                    aci "groton-ge-4/0/3.32" ari "paris" {
                        delay 5;
                        dynamic-profile ParisProfile;
                        routing-instance ParisRI;
                    }
```

```
                }
            }
            service empty {
                drop;
                agent-specifier {
                    aci "dunstable-ge-1/0/0.1" ari "kanata" {
                        dynamic-profile BasicPppoeProfile;
                        delay 10;
                    }
                }
            }
            service Premium {
                terminate;
                dynamic-profile PremiumProfile;
            }
            service Standard {
                terminate;
                max-sessions 10;
                dynamic-profile StandardProfile;
                agent-specifier {
                    aci "dunstable-ge-1/0/0.1" ari "kanata" {
                        dynamic-profile BasicPppoeProfile;
                        delay 10;
                    }
                }
            }
        }
    }
}
```

2. Configure the PPPoE underlying interface for the service name table.

```
interfaces {
    ge-2/0/0 {
        vlan-tagging;
        unit 1 {
            vlan-id 1;
            pppoe-underlying-options {
                dynamic-profile BasicPppoeProfile;
                service-name-table TableDynamicPPPoE;
            }
        }
```

```
            unit 2 {
                vlan-id 2;
                pppoe-underlying-options {
                    service-name-table TableDynamicPPPoE;
                }
            }
        }
    }
```

Table 11 on page 284 lists the service name, ACI value, and ARI value provided in several sample PPPoE client requests, and the name of the PPPoE underlying interface on which the router received each client request. The Results column describes the dynamic PPPoE subscriber interface created by the router based on *both* of the following:

- The values received from each PPPoE client during PPPoE negotiation
- The sequence in which the router evaluates the entries configured in the PPPoE service name table to find a match for the client's service name and ACI/ARI information, as described in "Evaluation Order for Matching Client Information in PPPoE Service Name Tables" on page 266

Table 11: Dynamic PPPoE Subscriber Interface Creation Based on PPPoE Client Request Values

PPPoE Client	Service Name	ACI Value	ARI Value	Receiving Underlying Interface	Results
Client 1	Premium	broadway-ge-1/0/1.1	london	ge-2/0/0.1	Matches ACI/ARI pair configured for any service. Router creates dynamic PPPoE subscriber interface over ge-2/0/0.1 using LondonProfile dynamic profile and LondonRI routing instance assigned to any service.
Client 2	Premium	dunstable-ge-1/0/1.0	toronto	ge-2/0/0.1	Matches base Premium service. Router creates dynamic PPPoE subscriber interface over ge-2/0/0.1 using PremiumProfile dynamic profile and routing instance associated with ge-2/0/0.1 underlying interface.

Table 11: Dynamic PPPoE Subscriber Interface Creation Based on PPPoE Client Request Values *(Continued)*

PPPoE Client	Service Name	ACI Value	ARI Value	Receiving Underlying Interface	Results
Client 3	empty	dunstable-ge-1/0/0.1	kanata	ge-2/0/0.1	Matches ACI/ARI pair configured for `empty` service and `Standard` service. Router creates dynamic PPPoE subscriber interface over `ge-2/0/0/.1` after a delay of 10 seconds. Router uses `BasicPPPoEProfile` dynamic profile and routing instance associated with `ge-2/0/0.1` underlying interface.
Client 4	empty	slinger-ge-1/0/0.1	chicago	ge-2/0/0.2	Because receiving underlying interface `ge-2/0/0.2` is *not* associated with a dynamic profile, router does not create a dynamic PPPoE subscriber interface, and drops any PADI or PADR control packets received from this client.
Client 5	Standard	slinger-ge-1/0/0.1	chicago	ge-2/0/0.1	Matches base `Standard` service. Router creates dynamic PPPoE subscriber interface over `ge-2/0/0.1` using `StandardProfile` dynamic profile and routing instance associated with `ge-2/0/0.1` underlying interface.

RELATED DOCUMENTATION

Evaluation Order for Matching Client Information in PPPoE Service Name Tables | 266

Subscriber Interfaces and PPPoE Overview | 186

Understanding PPPoE Service Name Tables | 260

Configuring PPPoE Service Name Tables | 269

Troubleshooting PPPoE Service Name Tables

IN THIS SECTION

Problem

Description

A misconfiguration of a PPPoE service name table can prevent PPPoE services from being properly activated. Configuration options for PPPoE service name tables are simple, which should simplify discovering where a misconfiguration exists. PPPoE clients cannot connect if the service name table contains no match for the service name tag carried in the PADI packet.

Symptoms

The symptom of a service name table misconfiguration is that the client connection process stops at the negotiation stage and the PADI packets are ignored. You can use the `show pppoe statistics` command to examine the PPPoE packet counts for a problem.

When the service name table is properly configured, packets sent and received increment symmetrically. The following sample output shows a PADO sent count equal to the PADI received count, and PADS sent count equal to the PADR received count. This output indicates that the PPPoE negotiation is proceeding successfully and that the service name table is not misconfigured.

```
user@host> show pppoe statistics ge-2/0/3.1

    Active PPPoE sessions: 2
    PacketType                        Sent         Received
      PADI                               0               16
      PADO                              16                0
```

```
    PADR                            0           16
    PADS                           16            0
    PADT                            0            0
    Service name error              0            0
    AC system error                 0            0
    Generic error                   0            0
    Malformed packets               0            0
    Unknown packets                 0            0
```

When the service name table is misconfigured, the output of the `show pppoe statistics` command indicates that the number of PADI packets received on the underlying interface is increasing, but the number of PADO packets sent remains at zero. The following sample output shows a PADI count of 100 and a PADO count of 0.

```
user@host> show pppoe statistics ge-2/0/3.1

Active PPPoE sessions: 0
    PacketType                   Sent     Received
    PADI                            0          100
    PADO                            0            0
    PADR                            0            0
    PADS                            0            0
    PADT                            0            0
    Service name error              0            0
    AC system error                 0            0
    Generic error                   0            0
    Malformed packets               0            0
    Unknown packets                 0            0
```

When you believe a misconfiguration exists, use the `monitor traffic` command on the underlying interface to determine which service name is being requested by the PPPoE client. The following sample output shows that the client is requesting Service1 in the service name tag.

```
user@host> monitor traffic interface ge-2/0/3.1 print-hex print-ascii
    Listening on ge-2/0/3.1, capture size 96 bytes

    11:49:41.436682  In PPPoE PADI [Service-Name "Service1"] [Host-Uniq UTF8] [TAG-0x120 UTF8]
[Vendor-Specific UTF8]
    0x0000   ffff ffff ffff 0090 1a42 0ac1 8100 029a     .........B......
    0x0010   8863 1109 0000 00c9 0101 0008 5365 7276     .c.........Serv
    0x0020   6963 6531 0103 0004 1200 9c43 0120 0002     ice1.......C....
```

```
0x0030   044a 0105 00ab 0000 0de9 0124 783a 3132     .J.........$x:12
0x0040   3030 3963                                   009c
```

You can then use the `show pppoe service-name-tables` command to determine whether you have misspelled the name of the service or perhaps not configured the service at all.

Cause

Typical misconfigurations appear in the service name table configurations.

Solution

Use the appropriate statements to correct the misconfiguration.

RELATED DOCUMENTATION

Configuring PPPoE Service Name Tables | 269

PPPoE Overview

Ethernet Interfaces User Guide for Routing Devices

CHAPTER 22

Changing the Behavior of PPPoE Control Packets

IN THIS CHAPTER

Enabling Advertisement of Named Services in PADO Control Packets

You can enable advertisement of named services in PADO control packets sent by the router to the PPPoE client to indicate the services that the router can offer. By default, advertisement of named services in PADO packets is disabled. You can enable PADO advertisement as a global option on the router when you configure the PPPoE protocol.

NOTE: Make sure the combined number and length of all named services advertised in the PADO packet does not exceed the MTU size of the PPPoE underlying interface.

To enable advertisement of named services in PADO packets:

- Configure the PPPoE protocol to enable PADO advertisement:

```
[edit protocols pppoe]
user@host# set pado-advertise
```

RELATED DOCUMENTATION

Understanding PPPoE Service Name Tables | 260

Configuring PPPoE Service Name Tables | 269

PPPoE Overview

Disabling the Sending of PPPoE Access Concentrator Tags in PADS Packets

By default, a router that functions as an access concentrator (AC) sends the AC-Name and AC-Cookie tags, along with the Service-Name, Host-Uniq, Relay-Session-Id, and PPP-Max-Payload tags, in the PPPoE Active Discovery Session (PADS) packet when it confirms a session with a PPPoE client. The AC-Name and AC-Cookie tags are defined as follows:

- AC-Name—String that uniquely identifies the particular AC
- AC-Cookie—Tag used by the AC to help protect against denial-of-service (DoS) attacks

If it is necessary for compatibility with your network equipment, you can prevent the router from sending the AC-Name and AC-Cookie tags in the PADS packet.

To prevent the router from transmitting the AC-Name and AC-Cookie tags in the PADS messages:

- Specify that PADS messages with AC-Name and AC-Cookie tags are not sent.

  ```
  [edit protocols pppoe]
  user@host# set no-send-pads-ac-info
  ```

  The `no-send-pads-ac-info` statement affects PADS packets sent only on PPPoE interfaces configured on the router after you configure this statement. It has no effect on PADS packets sent on previously created PPPoE interfaces.

RELATED DOCUMENTATION

PPPoE Overview

Discarding PADR Messages to Accommodate Abnormal CPE Behavior

This topic describes how to avoid a situation where certain CPEs respond inappropriately to normal router behavior.

During PPPoE session negotiation, the router returns PADS messages in response to PADR messages when it accepts or rejects the PPPoE session. The router adds an error tag to the PADS message when it detects a problem.

AC-System-Error is one such tag. This tag is inserted when the router imposes automatic throttling in response to excessive CPU consumption, excessive subscriber connections, or physical interfaces cycling up and down.

When the CPE receives a PADS message with this tag, the typical behavior is to retry sending PADR messages to the router or to restart session negotiation by sending PADI messages. However, some CPEs may respond inappropriately with the result that their subscribers are never connected until the CPE is rebooted.

To avoid this situation when such CPEs have access to your network, you can configure the router to silently discard PADR messages in situations where the PADS would include the AC-System-Error tag. The consequence is that the CPE resends PADR messages. When the conditions that result in the AC-System-Error tag are no longer present, the router once again evaluates PADR packets to determine whether to accept or reject the session.

To silently discard PADR packets:

- Specify that PADS messages with AC-System-Error tags are not sent.

```
[edit protocols pppoe]
user@host# set no-send-pads-error
```

RELATED DOCUMENTATION

PPPoE Overview

CHAPTER 23

Monitoring and Managing Dynamic PPPoE for Subscriber Access

IN THIS CHAPTER

Verifying and Managing Dynamic PPPoE Configuration

IN THIS SECTION

Purpose

View or clear information about dynamic PPPoE logical interfaces, underlying interfaces for dynamic PPPoE logical interfaces, and PPPoE statistics.

Action

- To display information about the properties of all PPPoE underlying interfaces associated with a dynamic PPPoE profile:

```
user@host> show pppoe underlying-interfaces
```

- To display information about the PPPoE properties of a specified underlying interface associated with a dynamic PPPoE profile:

```
user@host> show pppoe underlying-interfaces interface-name
```

- To display session-specific information about PPPoE interfaces, including whether the interface was dynamically created or statically created:

```
user@host> show pppoe interfaces
```

- To display information for a specified PPPoE service name table, including the assigned dynamic profile and routing instance, if configured:

```
user@ host> show pppoe service-name-tables table-name
```

- To display information about all active PPPoE sessions on the router:

```
user@host > show pppoe sessions
```

- To display information for all active PPPoE sessions established for a specified service name:

```
user@host > show pppoe sessions service service-name
```

- To display information for all active PPPoE sessions established for a specified agent circuit identifier (ACI) or agent remote identifier (ARI) string:

```
user@host > show pppoe sessions aci “west-ge-2/0/3”
user@host > show pppoe sessions ari “sunnyvale”
```

- To display PPPoE control packet statistics for all PPPoE sessions:

```
user@host> show pppoe statistics
```

- To display PPPoE control packet statistics for a specified PPPoE underlying interface:

```
user@host> show pppoe statistics interface-name
```

- To clear (reset) PPPoE control packet statistics for all PPPoE sessions:

```
user@host> clear pppoe statistics
```

- To clear (reset) PPPoE control packet statistics for a specified underlying Ethernet interface:

```
user@host> clear pppoe statistics underlying-interface-name
```

- To display summary information about PPPoE subscriber sessions currently undergoing lockout or currently in a lockout grace period on all PPPoE underlying interfaces:

```
user@host> show pppoe lockout
```

- To display summary information about PPPoE subscriber sessions currently undergoing lockout or currently in a lockout grace period on the specified PPPoE underlying interface:

```
user@host> show pppoe lockout underlying-interface-name
```

- To display information about the lockout condition or lockout grace period for all PPPoE subscriber sessions associated with the specified ATM encapsulation type identifiers:

```
user@host> show pppoe lockout atm-identifier device-name device-name vpi vpi-identifier vci vci-
identifier
```

- To display information about the lockout condition or lockout grace period for all PPPoE subscriber sessions associated with the specified VLAN encapsulation type identifiers:

```
user@host> show pppoe lockout vlan-identifier device-name device-name svlan-id svlan-
identifier vlan-id vlan-identifier
```

RELATED DOCUMENTATION

CLI Explorer

4

PART

Configuring MLPPP for Subscriber Access

CHAPTER 24

MLPPP Support for LNS and PPPoE Subscribers Overview

IN THIS CHAPTER

MLPPP Overview

IN THIS SECTION

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.

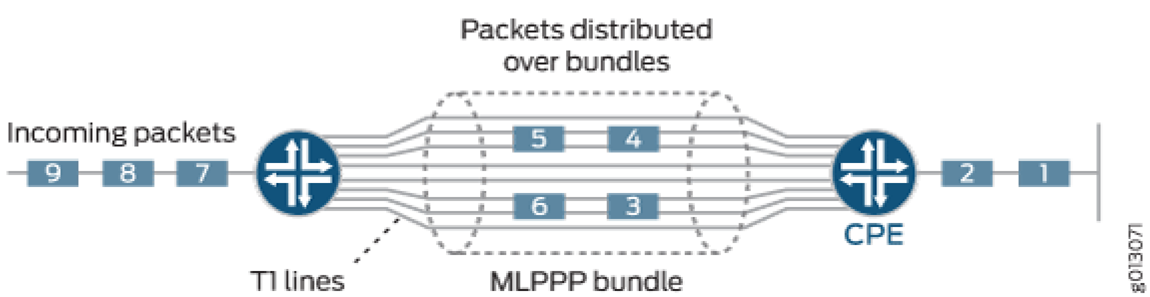
Packets received with an MLPPP header are subject to fragmentation, reassembly, and sequencing. Packets received without the MLPPP header cannot be sequenced and can be delivered only on a first-come, first-served basis.

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 5 on page 298 shows how incoming packets are distributed and aggregated into an MLPPP bundle.

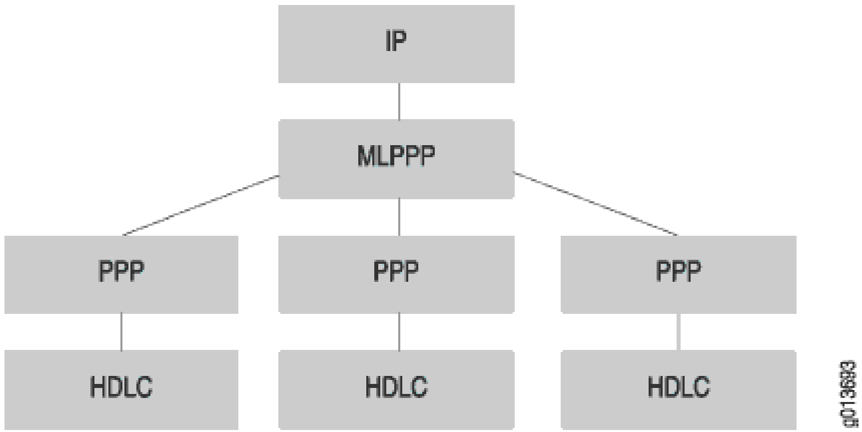
Figure 5: MLPPP Aggregation of Traffic Into Single Bundle



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 6 on page 299 illustrates interface stacking with MLPPP.

Figure 6: 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.

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.

Change History for MLPPP

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Table 12: Change History for MLPPP

Release	Description
14.1	MLPPP for subscriber access is supported starting in Junos OS Release 14.1.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	MLPPP for subscriber access is supported starting in Junos OS Release 14.1.

RELATED DOCUMENTATION

MLPPP Support for LNS and PPPoE Subscribers Overview | 300
Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series | 304
Understanding MLPPP Link Fragmentation and Interleaving | 307

MLPPP Support for LNS and PPPoE Subscribers Overview

IN THIS SECTION

Starting in Junos OS Release 14.1, 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 router 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.

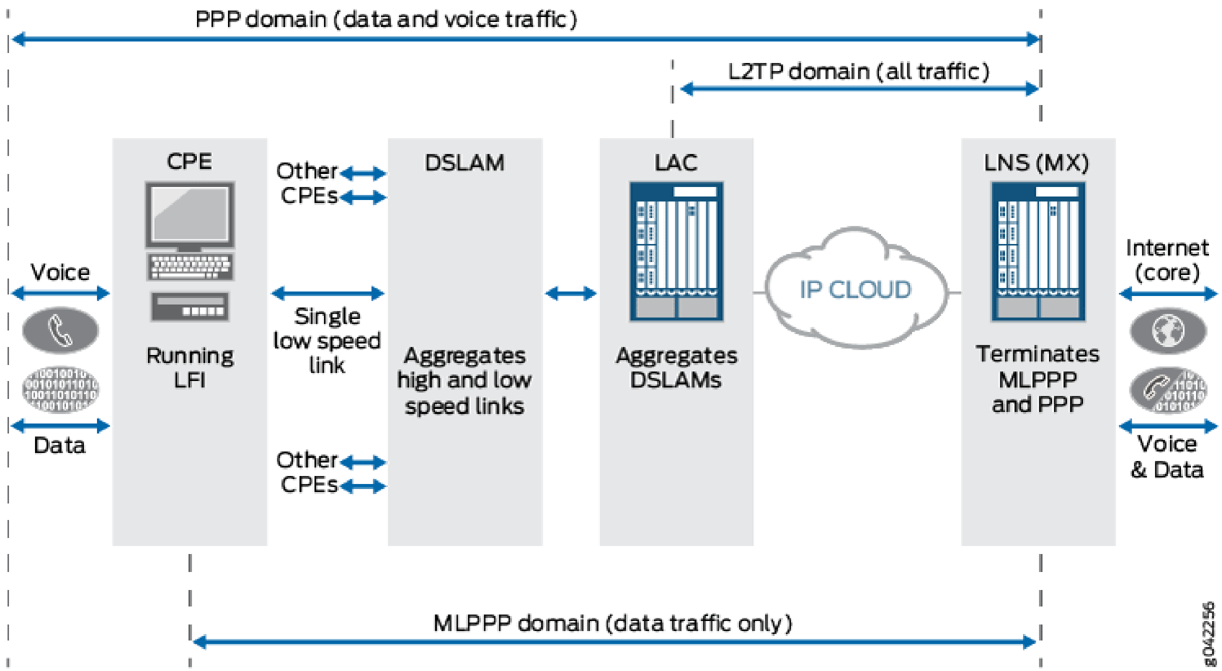
Each bundle accepts only one member link. If more than one member link attempts to join the same bundle, the system fails the new member session.

Dual-stack is supported for the bundle.

LNS Subscribers and MX Series

Figure 7 on page 302 shows a network diagram with the MX Series functioning as the LNS. Both PPP and MLPPP bundles are terminated at the LNS.

Figure 7: MLPPP Bundles Terminated at MX Series as the LNS Network



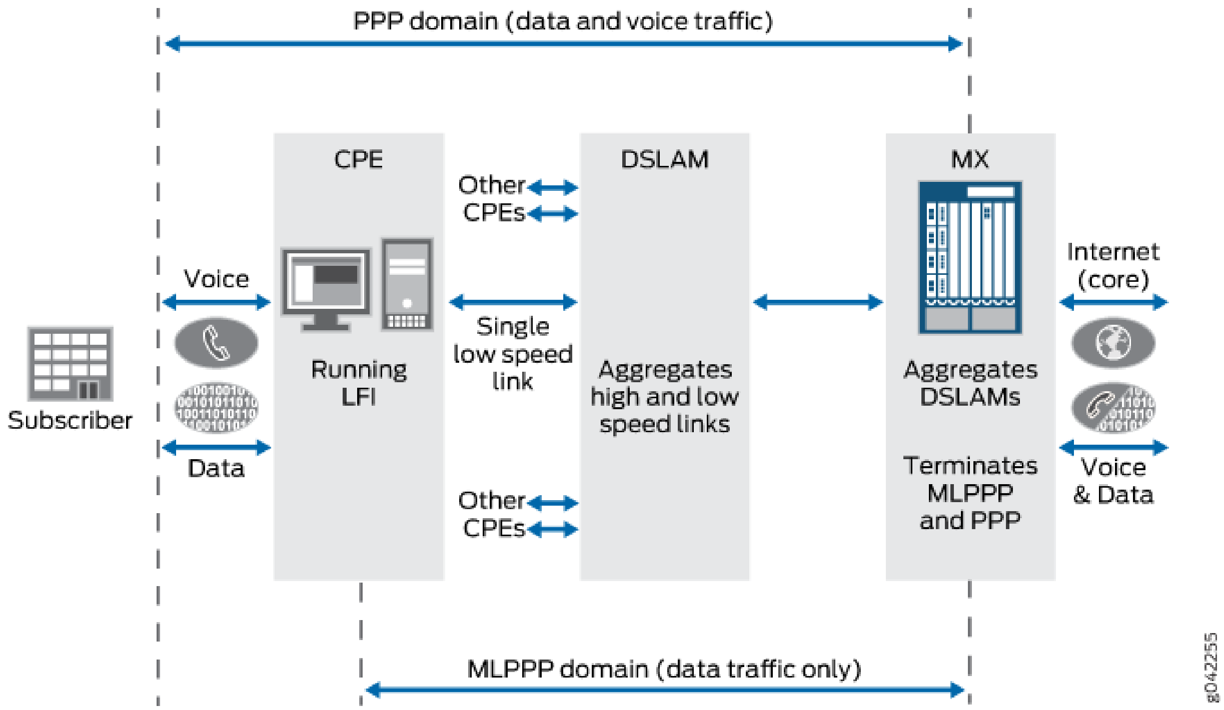
The following three domains are shown passing traffic through the LNS network:

- 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 8 on page 303 shows a network diagram with the MX Series terminating PPPoE sessions that include both the PPP and MLPPP bundles.

Figure 8: 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

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, 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.

RELATED DOCUMENTATION

MLPPP Overview | 297

Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series | 304

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

Starting in Junos OS Release 14.1, subscribers on MX Series router to multilink PPP (MLPPP) for L2TP network server (LNS) or to Point-to-Point Protocol over Ethernet (PPPoE, terminated and tunneled) can access a variety of new features.

- 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).

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1	Starting in Junos OS Release 14.1, subscribers on MX Series router to multilink PPP (MLPPP) for L2TP network server (LNS) or to Point-to-Point Protocol over Ethernet (PPPoE, terminated and tunneled) can access a variety of new features.

RELATED DOCUMENTATION

MLPPP Support for LNS and PPPoE Subscribers Overview | 300
Mixed Mode Support for MLPPP and PPP Subscribers Overview | 305
MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

Mixed Mode Support for MLPPP and PPP Subscribers Overview

IN THIS SECTION

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 Overview | 300

Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series | 304

Configuring L2TP Client Access to Support MLPPP for Static Subscribers | 326

Example: Configuring Dynamic LNS MLPPP Subscribers | 363

CHAPTER 25

Configuring MLPPP Link Fragmentation and Interleaving

IN THIS CHAPTER

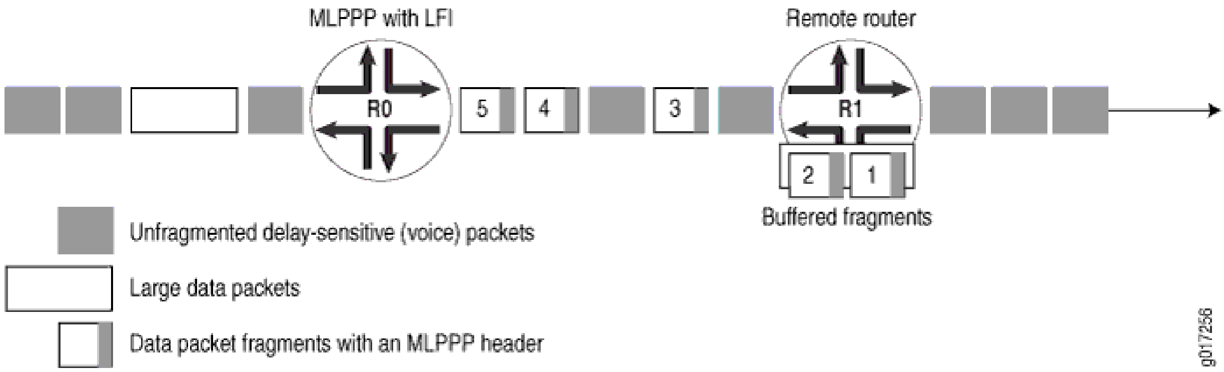
Understanding MLPPP Link Fragmentation and Interleaving

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 9 on page 308 shows how LFI processes packets.

Figure 9: 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 | 309
Understanding Fragmented Packet Queuing | 312
Understanding Sequenced Packet Fragment Drops | 316

Understanding MLPPP and Fragmentation-Maps

IN THIS SECTION

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;
            }
        }
    }
}
```

NOTE: The per-forwarding class `drop-timeout` statement enabling you to change the resequencing interval in milliseconds for each fragmentation class is not supported in the fragmentation map.

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.

    NOTE: The per-forwarding class `multilink-class` statement enabling you to map a forwarding class into a multiclass MLPPP is not supported for `si` MLPPP bundles.

- (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

Understanding MLPPP Link Fragmentation and Interleaving | 307
Understanding Fragmented Packet Queuing | 312
Understanding Sequenced Packet Fragment Drops | 316

Understanding Fragmented Packet Queuing

IN THIS SECTION

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.

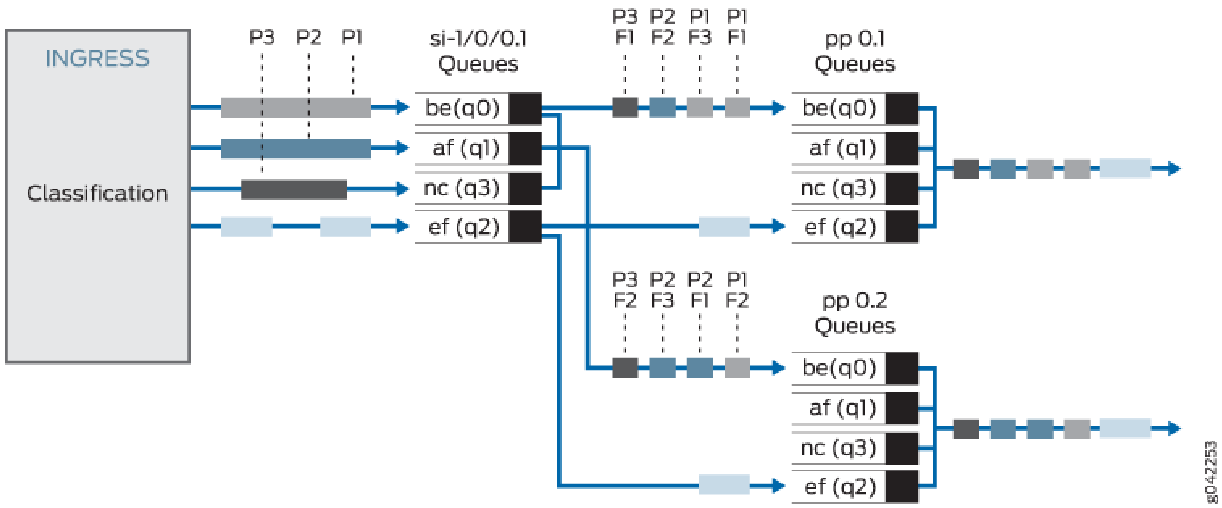
- Traffic flows of a forwarding class that has MLPPP fragmentation configured are distributed from the inline services `si` bundle interface queues to the member link queues (queue 0) following a round-robin method.
- Traffic flows of a forwarding class without MLPPP fragmentation are distributed from 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 10 on page 313 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 10: 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.

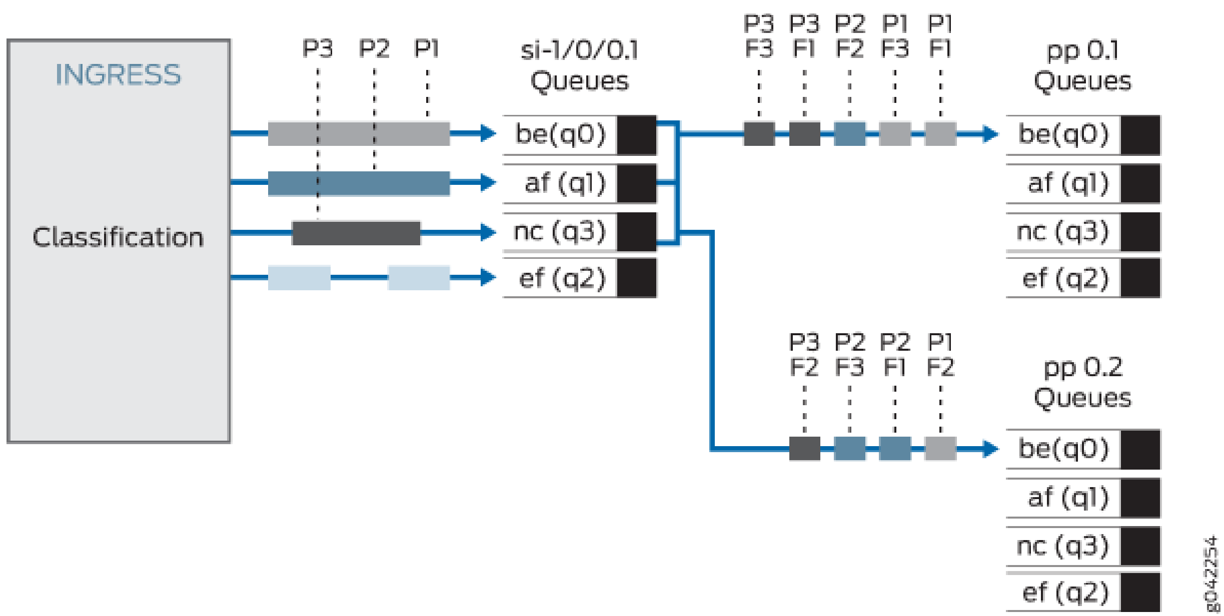
This section contains the following topics:

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 11 on page 314 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 11: 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.

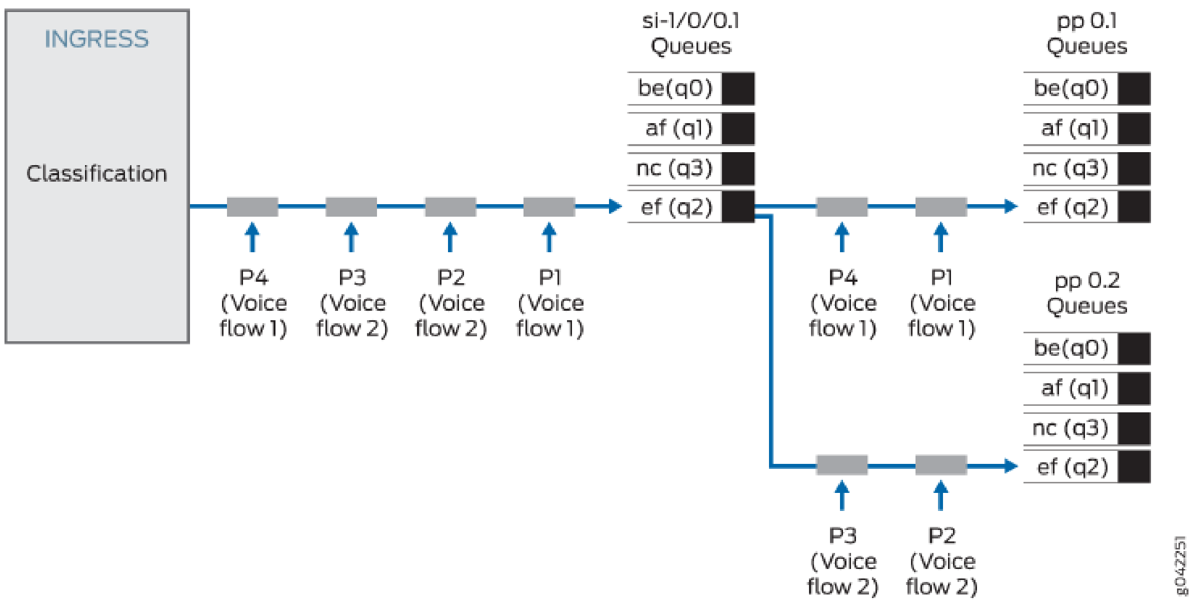
After exiting the `si` interface, Microcode adds a header of approximately 40 bytes to the MLPPP packets. When configuring the class-of-service shaping, you may need to adjust bytes to account for this.

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 12 on page 315 shows how LFI packet queuing is performed on the member links.

Figure 12: 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 | 307
Understanding MLPPP and Fragmentation-Maps | 309
Understanding Sequenced Packet Fragment Drops | 316

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.

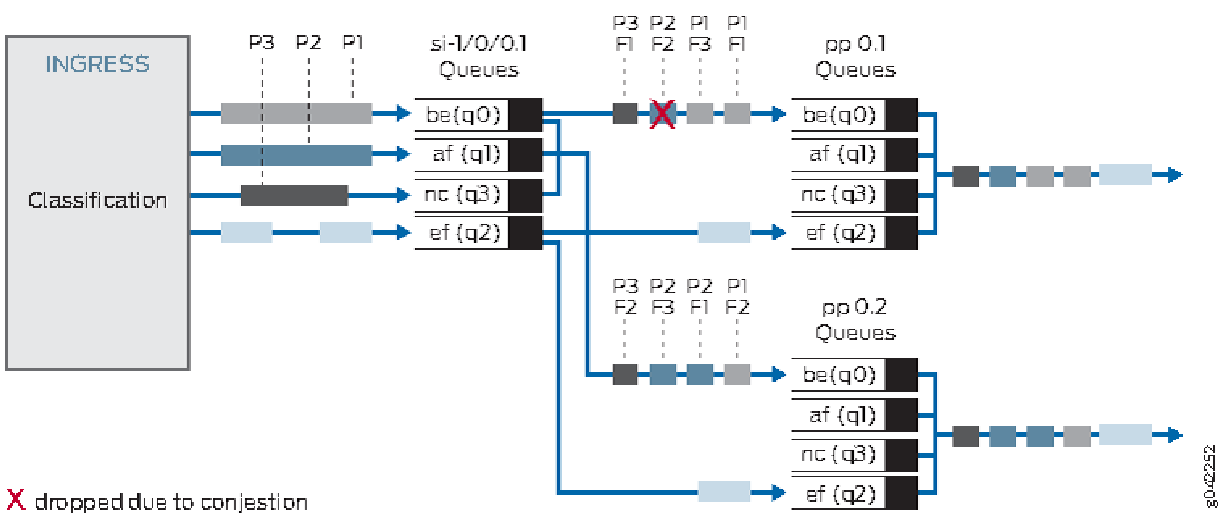
The MX Series provides two stages of queuing for packets exiting an MLPPP bundle:

- 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 13 on page 316. Packet flows in the figure use the notation P*x*,F*x*; for example, P1,F1 represents Packet 1, Fragment 1.

Figure 13: 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 14 on page 317 and Figure 15 on page 318 for member link scheduler hierarchies.

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

Figure 14: si Bundle Interface Scheduler Hierarchy

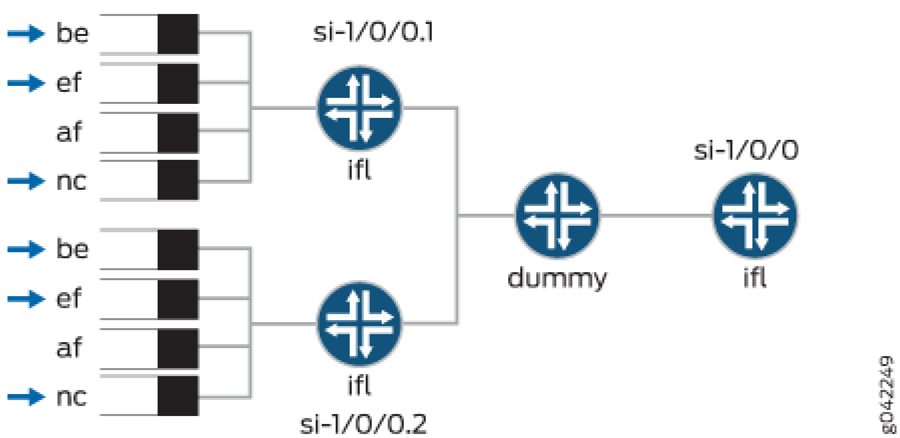
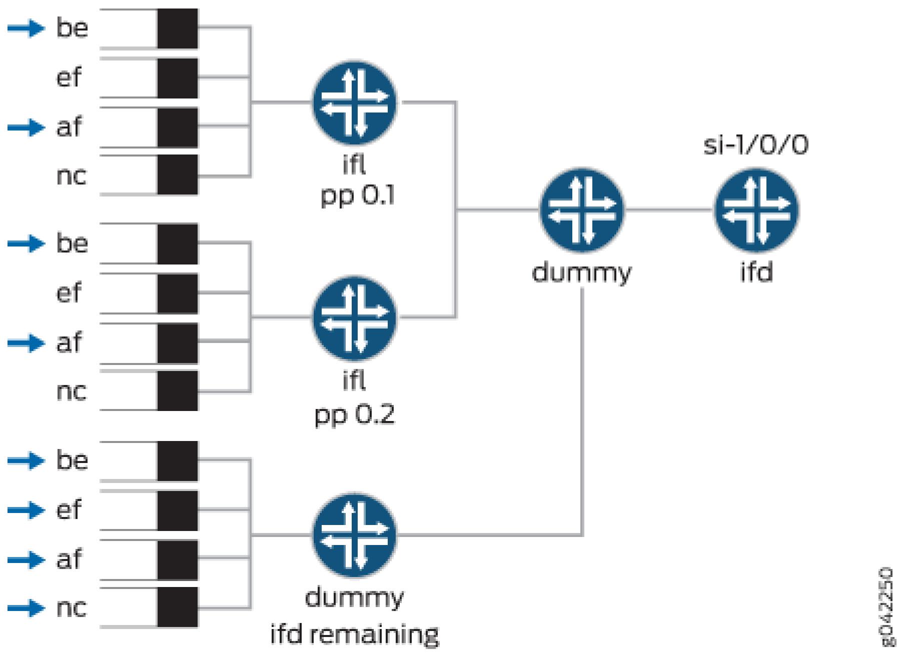


Figure 15: MLPPP Member Link Scheduler Hierarchy



RELATED DOCUMENTATION

Understanding MLPPP Link Fragmentation and Interleaving | 307

Understanding MLPPP and Fragmentation-Maps | 309

Understanding Fragmented Packet Queuing | 312

CHAPTER 26

Configuring Inline Service Interfaces for LNS and PPPoE Subscribers

IN THIS CHAPTER

MLPPP Bundles and Inline Service Logical Interfaces Overview

IN THIS SECTION

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

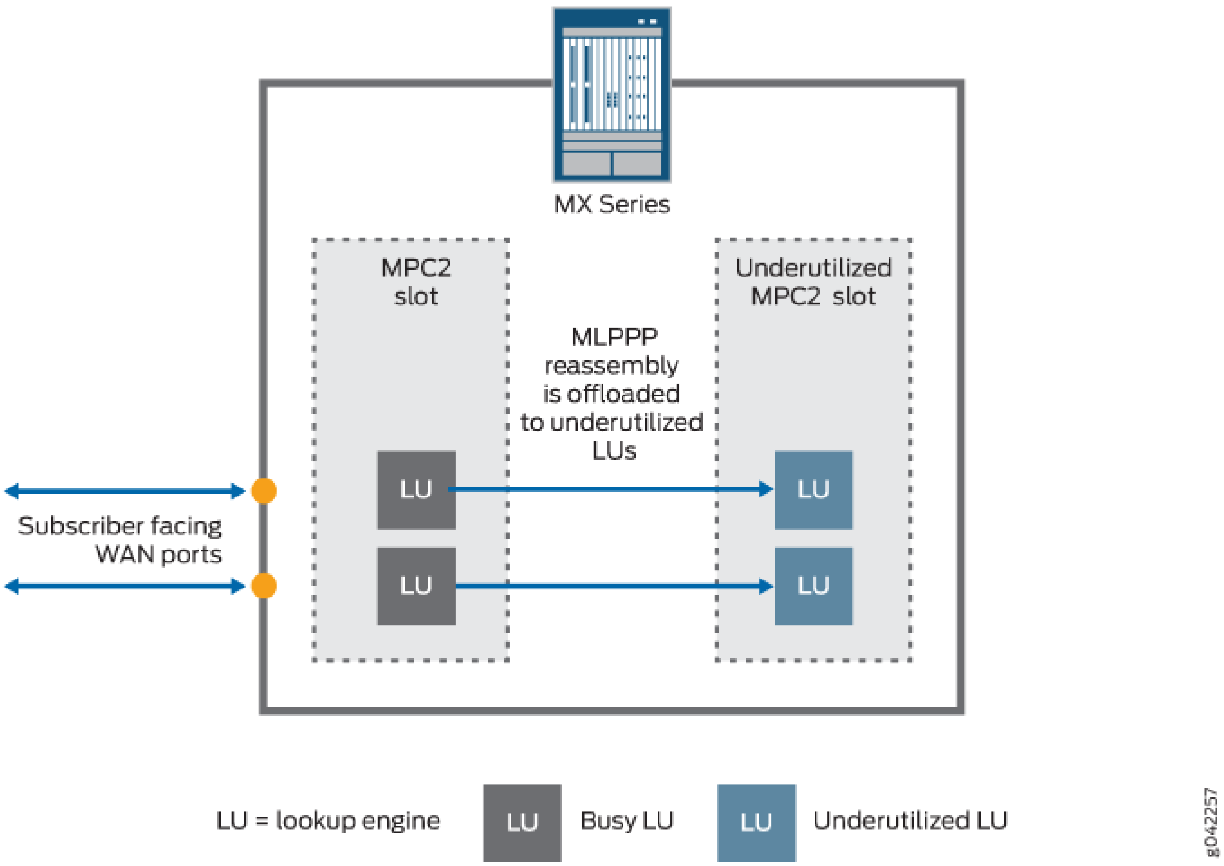
This topic contains the following sections:

Distribution of Reassembly Processing

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 16 on page 320 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 16: 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 | 321
Configuring Inline Service Interface for PPPoE and LNS Subscribers | 322
Understanding MLPPP Link Fragmentation and Interleaving | 307

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 MPC1-3D 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 | 322
Configuring Service Device Pools for Load Balancing PPPoE and LNS Subscribers | 324
MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

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 | 324

MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

Enabling Inline Service Interfaces for PPPoE and LNS Subscribers | 321

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 321.

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 | 322

MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

Example: Configuring Dynamic LNS MLPPP Subscribers | 363

CHAPTER 27

Configuring L2TP Access Client for MLPPP Subscribers

IN THIS CHAPTER

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/2lP";
                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 | 305
MLPPP Support for LNS and PPPoE Subscribers Overview | 300
Example: Configuring Static LNS MLPPP Subscribers | 332

Configuring L2TP Client Access to Support MLPPP for Dynamic Subscribers

To enable support for MLPPP over LT2P network server (LNS) 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 | **363**

MLPPP Support for LNS and PPPoE Subscribers Overview | **300**

CHAPTER 28

Configuring Static MLPPP Subscribers for MX Series

IN THIS CHAPTER

Example: Configuring Static LNS MLPPP Subscribers

IN THIS SECTION

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 321.

- Configured the inline service (si) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 322.

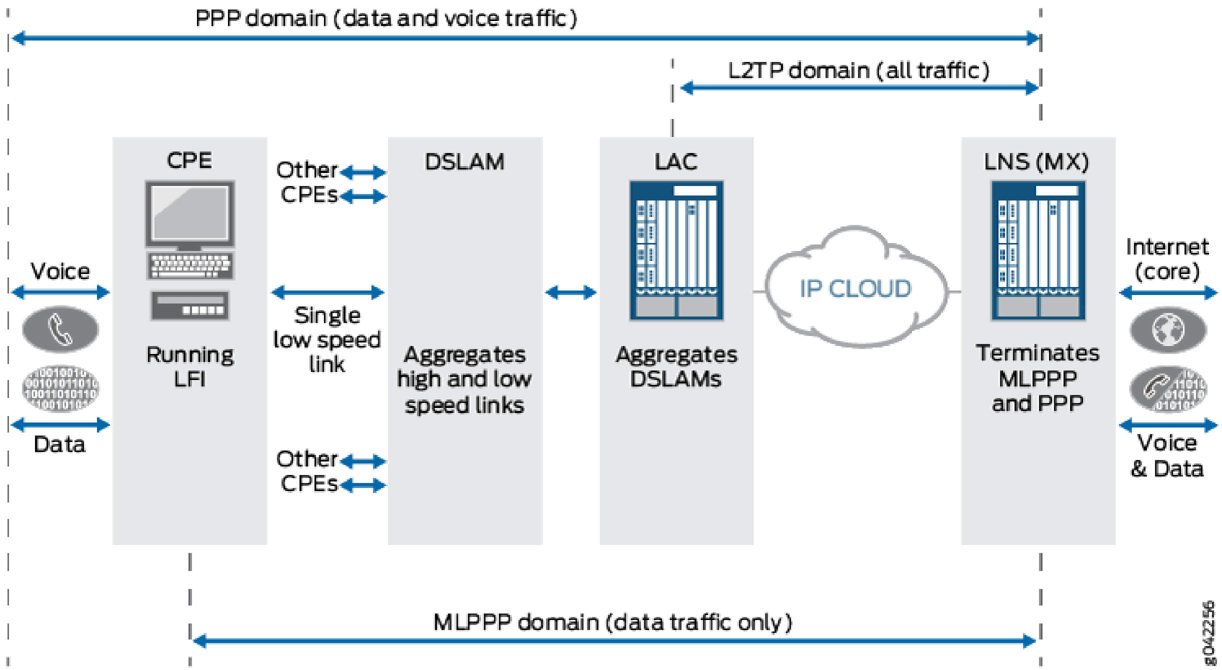
Overview

IN THIS SECTION

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 17 on page 334 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 17: 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

IN THIS SECTION

To configure static L2TP network server (LNS) multilink (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-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 10.1.1.2
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 10.1.1.2
```

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 10.1.1.2;
        }
        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

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Inline Services Interface Information

Purpose

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

Action

```
user@host> 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

```
user@host> 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

```
user@host> 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

```
user@host> show subscribers extensive

  Type: L2TP
User Name: user@test.com
IP Address: 10.80.80.10
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: user@test.com
IP Address: 10.80.80.10
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 Overview | 300
Configuring L2TP Client Access to Support MLPPP for Static Subscribers | 326
Example: Configuring Static PPPoE MLPPP Subscribers | 347

Example: Configuring Static PPPoE MLPPP Subscribers

IN THIS SECTION

This example shows how to configure static Point-to-Point Protocol over Ethernet (PPPoE) MLPPP for terminated and tunneled 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 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 321.
- Configured the inline service (`si`) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 322.

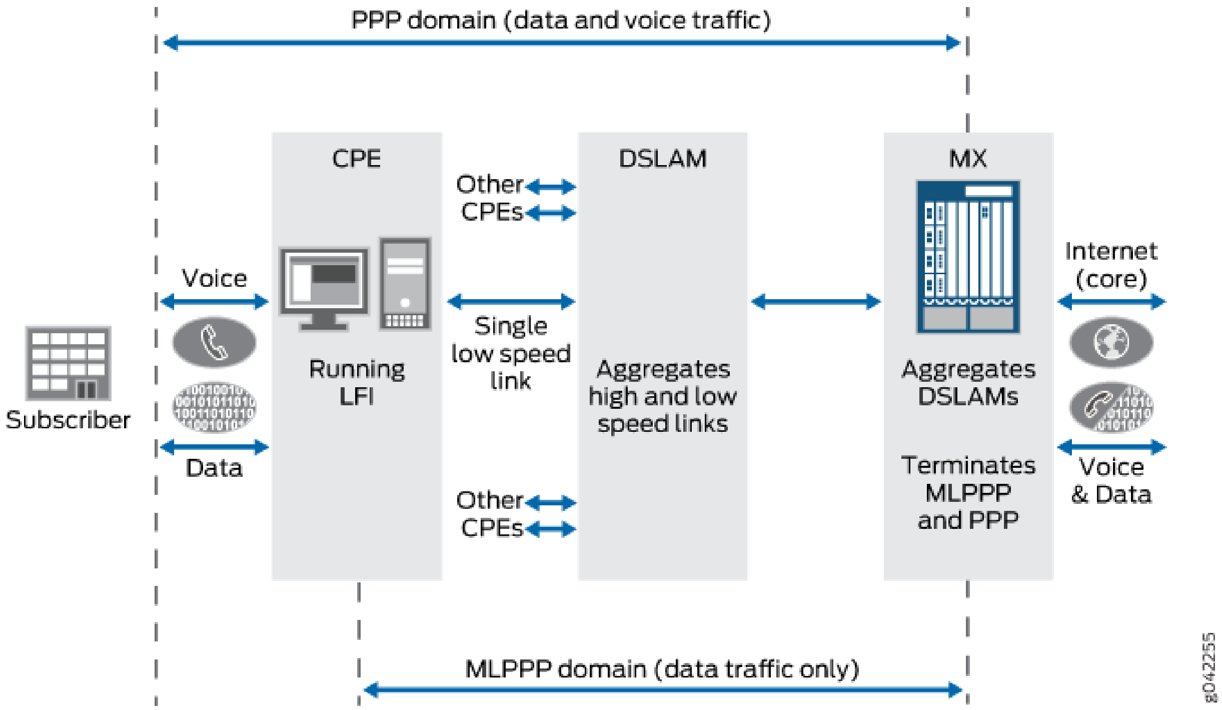
Overview

IN THIS SECTION

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 18 on page 349 shows how the different types of traffic traverse through a network where the MX Series terminates PPPoE sessions.

Topology

Figure 18: 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

IN THIS SECTION

To configure static PPPoE MLPPP for terminated and tunneled 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 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 2001:db8:204::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.

Step-by-Step Procedure

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 2001:db8:204::1:1:2/64
```

5. Configure the second of three logical interfaces.

Step-by-Step Procedure

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.

Step-by-Step Procedure

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 2001:db8:204::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

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Bundle IFL Information

Purpose

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

Action

```
user@host> 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: 10.80.80.1, 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

```
user@host> 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: 00:00:5e:00:53:42,
      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@host> 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: 00:00:5e:00:53:32
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 Overview | 300

MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

Example: Configuring Dynamic PPPoE MLPPP Subscribers | 386

CHAPTER 29

Configuring Dynamic MLPPP Subscribers for MX Series

IN THIS CHAPTER

Example: Configuring Dynamic LNS MLPPP Subscribers

IN THIS SECTION

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 321.
- Configured the inline service (si) interface for LNS subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 322.
- 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 324.

Overview

IN THIS SECTION

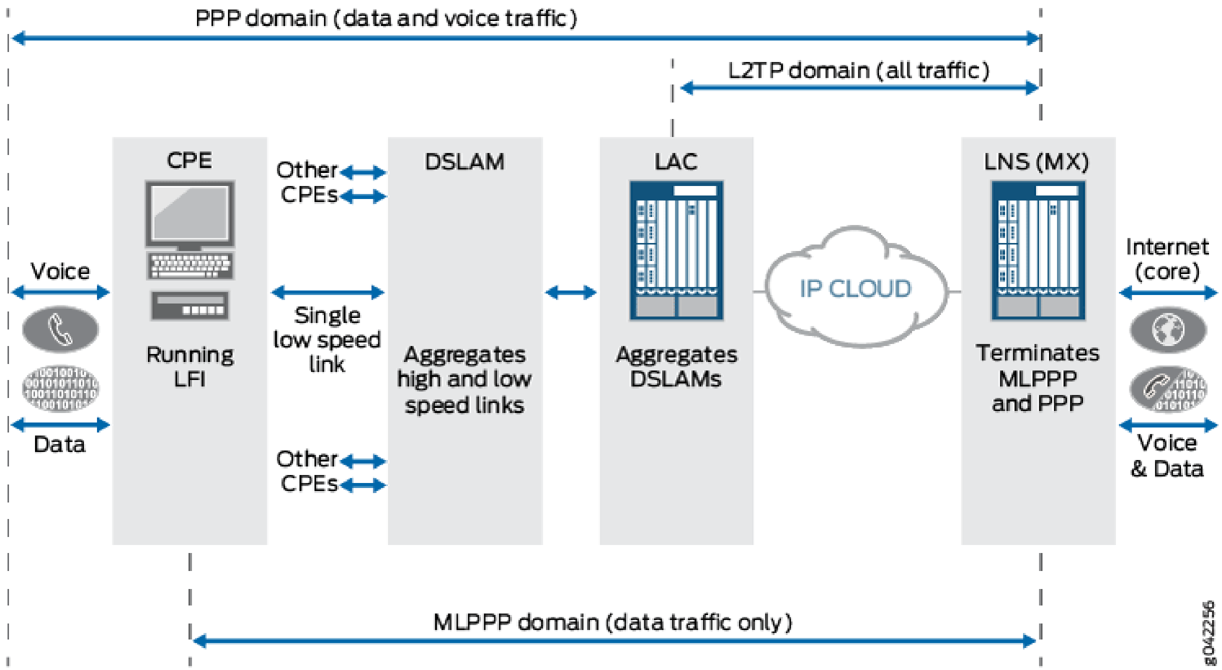
An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For dynamic LNS MLPPP subscribers, you configure the dynamic member link IFLs using dynamic profiles. The member link dynamic profile includes the `family mlppp` statement containing the bundle dynamic profile 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 19 on page 365 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 19: 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

IN THIS SECTION

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 10.1.1.1
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 10.1.1.1**

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 368.

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 372 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 368.

   `[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 366 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 368.

   ```
   [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 370 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 10.1.1.1;
            }
```

```
            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

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Subscriber Information

Purpose

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

Action

```
user@host> show subscribers extensive
Type: L2TP
User Name: lns-client
IP Address: 198.51.100.20
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: 198.51.100.20
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

```
user@host> 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: 00:00:5e:00:53:72,
      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: 10.0.0.1, 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

```
user@host> 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: 00:00:5e:00:53:82,
      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 Overview | 300

Mixed Mode Support for MLPPP and PPP Subscribers Overview | 305

Configuring L2TP Client Access to Support MLPPP for Dynamic Subscribers | 329

Example: Configuring Dynamic PPPoE MLPPP Subscribers

IN THIS SECTION

This example shows how to configure dynamic Point-to-Point Protocol over Ethernet (PPPoE) 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 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 321.
- Configured the inline service (si) interface for PPPoE subscribers. See "Configuring Inline Service Interface for PPPoE and LNS Subscribers" on page 322.
- 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 324.

Overview

IN THIS SECTION

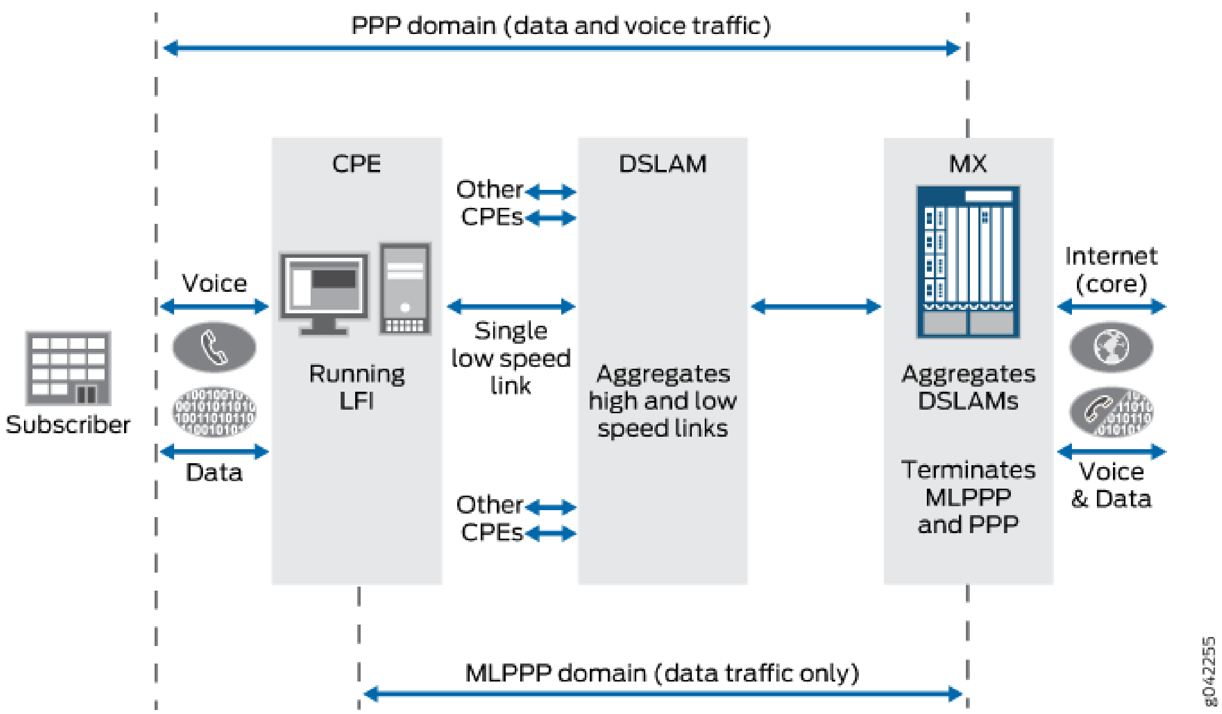
An MLPPP subscriber consists of two IFLs (logical interfaces), a member link, and a bundle. For dynamic PPPoE MLPPP subscribers, you configure the dynamic `pp0` member link IFLs using dynamic profiles. The `pp0` member link dynamic profile includes the `family mlppp` statement containing the dynamic 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 20 on page 387 shows how the different types of traffic traverse through a network where the MX Series terminates PPPoE sessions.

Topology

Figure 20: 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

IN THIS SECTION

To configure dynamic PPPoE 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 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 392 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 388 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 390, 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 390, 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

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Subscriber Information

Purpose

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

Action

```
user@host> show subscribers extensive
Type: PPPoE
User Name: dual-stack-v4v6-user@example.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: 00:00:5E:00:53:02
State: Active
PPP State: Tunneled
Local IP Address: 198.51.100.21
Remote IP Address: 198.51.100.22
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

```
user@host> 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: 00:00:5e:00:53:72,
      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: 198.51.100.11, 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

```
user@host> 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: 00:00:5e:00:53:82,
      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 Overview | 300
Mixed Mode Support for MLPPP and PPP Subscribers Overview | 305
MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

CHAPTER 30

Configuring Dynamic PPP Subscriber Services

IN THIS CHAPTER

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 407. 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 | 407
Configuring PPP Subscriber Services for MLPPP Bundles | 407

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:

- MX Series 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 | 406

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 408.
2. Attach a dynamic profile to the MLPPP bundle interface.

See "Attaching Dynamic Profiles to MLPPP Bundles" on page 409.

RELATED DOCUMENTATION

Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces | 407
Example: Minimum MLPPP Dynamic Profile | 409
Example: Configuring CoS on Static LSQ MLPPP Bundle Interfaces | 410

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 407.

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

Hardware Requirements for PPP Subscriber Services on Non-Ethernet Interfaces | 407
Configuring PPP Subscriber Services for MLPPP Bundles | 407

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 | 407
Configuring PPP Subscriber Services for MLPPP Bundles | 407
Dynamic Profiles Overview
Configuring PPP Subscriber Services for MLPPP Bundles | 407
Example: Minimum MLPPP Dynamic Profile | 409
Example: Configuring CoS on Static LSQ MLPPP Bundle Interfaces | 410

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

Attaching Dynamic Profiles to MLPPP Bundles | 409

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 | 407

Layer 2 Service Package Capabilities and Interfaces

CHAPTER 31

Monitoring and Managing MLPPP for Subscriber Access

IN THIS CHAPTER

MLPPP Subscriber Accounting Statistics Overview

IN THIS SECTION

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-VSAs 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.

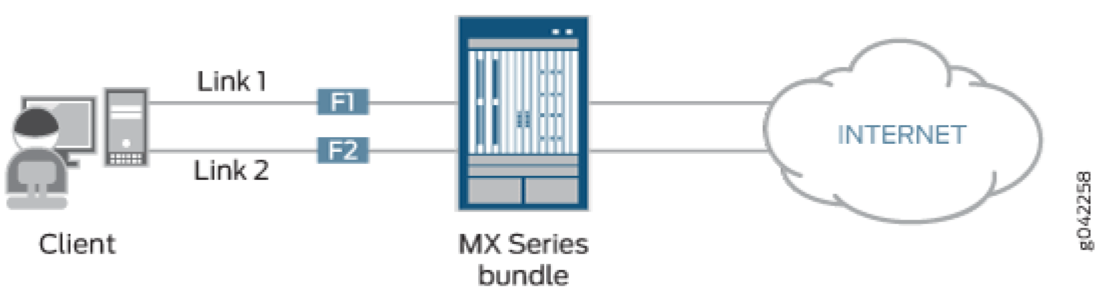
The following topics describe the statistics collection process in the lookup engine for member links and its bundle.

Member Link and Bundle Statistics Collection

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 21 on page 416 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 21: 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 21 on page 416).

- 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 = "user@example.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 = 10.0.0.3
        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 = 10.1.1.2
        Acct-Unique-Session-Id = "03eeef735aef3520"
        Timestamp = 1350604541
        Request-Authenticator = Verified
```

RELATED DOCUMENTATION

MLPPP Bundles and Inline Service Logical Interfaces Overview | 319

MLPPP Support for LNS and PPPoE Subscribers Overview | 300

Supported Features for MLPPP LNS and PPPoE Subscribers on the MX Series | 304

5

PART

Configuring ATM for Subscriber Access

CHAPTER 32

Configuring ATM to Deliver Subscriber-Based Services

IN THIS CHAPTER

ATM for Subscriber Access Overview

IN THIS SECTION

By using the ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) and a supported Modular Port Concentrator (MPC), you can configure the MX Series router to support configurations that enable subscribers to access the router over an ATM network using ATM Adaptation

Layer 5 (AAL5) permanent virtual connections (PVCs). Using these configurations enables the delivery of subscriber-based services, such as *class of service* (CoS) and firewall filters, for subscribers accessing the router over an ATM network.

Supported Configurations for ATM Subscriber Access

On MX Series routers with MPC/MIC interfaces that use the ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM), you can create the following configurations to enable subscribers to access the router over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual connections (PVCs):

- PPP-over-Ethernet-over-ATM
- Routed IP-over-ATM
- Bridged IP-over-Ethernet-over-ATM
- PPP-over-ATM
- Concurrent PPP-over-Ethernet-over-ATM interfaces and IP-over-Ethernet-over-ATM interfaces on a single ATM PVC

PPP-over-Ethernet-over-ATM Configurations

PPP-over-Ethernet-over-ATM (PPPoE-over-ATM) configurations support both statically created and dynamically created PPPoE (`pp0`) logical subscriber interfaces over static ATM underlying interfaces. Most PPPoE and subscriber services features supported on terminated connections and tunneled (L2TP access concentrator, or LAC) connections are also supported for access to an MX Series router over an ATM network.

PPPoE-over-ATM configurations require static configuration of the underlying ATM physical interface and ATM *logical interface*. You can configure the PPPoE (`pp0`) subscriber interface either dynamically, by means of a dynamic profile, or statically. You must also configure the ATM underlying interface with PPPoE-over-ATM logical link control (LLC) encapsulation (`encapsulation ppp-over-ether-over-atm-llc`).

Using dynamic PPPoE-over-ATM configurations for ATM subscriber access enables you to configure an MX Series router to dynamically create PPPoE logical subscriber interfaces over static ATM underlying interfaces only when needed; that is, when a subscriber logs in on the associated underlying interface. Dynamic PPPoE over static ATM configurations are *not* supported on M Series routers and T Series routers.

Optionally, you can dynamically or statically apply subscriber services such as class of service (CoS) and firewall filters to the PPPoE (`pp0`) subscriber interface. For PPPoE-over-ATM configurations that create a dynamic PPPoE subscriber interface, you can configure CoS attributes and firewall filters in the dynamic profile that defines the `pp0` subscriber interface. For PPPoE-over-ATM configurations that create a static

PPPoE subscriber interface, you can statically configure CoS attributes and firewall filters as you would for any static interface configured on an MX Series router.

Routed IP-over-ATM Configurations

Routed IP-over-ATM (IPoA) configurations support statically created IPv4 and IPv6 logical subscriber interfaces over static ATM underlying interfaces. IPoA configurations are typically used to implement business digital subscriber line (DSL) connections that do not require connection negotiation for address assignment.

IPoA configurations require static configuration of the ATM underlying interface, IPv4 interface, IPv6 interface, CoS attributes, and firewall fiters. Dynamic configuration of these components is not supported.

To configure IPoA subscriber access, specify either of the following encapsulation types on the ATM underlying interface:

- For IPoA encapsulation with logical link control (LLC), configure ATM subnetwork attachment point (SNAP) encapsulation (`encapsulation atm-snap`).
- For IPoA encapsulation with virtual circuit (VC) multiplexing, configure ATM VC multiplex encapsulation (`encapsulation atm-vc-mux`).

Optionally, you can statically configure subscriber services such as CoS and firewall filters and apply them to the IPv4 or IPv6 interface; you *cannot* use a dynamic profile for this purpose.

Bridged IP-over-Ethernet-over-ATM Configurations

Bridged IP-over-Ethernet-over-ATM (IPoE-over-ATM) configurations support statically created IPv4 and IPv6 logical subscriber interfaces over static ATM underlying interfaces. Like IPoA configurations, IPoE-over-ATM configurations are typically used in topologies that do not require connection negotiation for address assignment.

For bridged IP-over-Ethernet-over-ATM configurations on an MX Series router, you must configure the ATM underlying interface with Ethernet-over-ATM LLC encapsulation (`encapsulation ether-over-atm-llc`).

IPoE-over-ATM configurations require static configuration of the ATM underlying interface, IP interface, CoS attributes, and firewall fiters. Dynamic configuration of these components is not supported. Optionally, you can statically configure subscriber services such as class of service (CoS) and firewall filters and apply them to the IPv4 or IPv6 interface; you *cannot* use a dynamic profile for this purpose.

PPP-over-ATM Configurations

PPP-over-ATM (PPPoA) configurations support statically created PPP logical subscriber interfaces over static ATM underlying interfaces. Most features supported for PPPoE configurations are also supported for PPP access to an MX Series router over an ATM network.

PPPoA configurations require static configuration of the ATM underlying interface and PPP subscriber interface.

To configure PPPoA subscriber access, you must configure either of the following encapsulation types on each PPP logical subscriber interface:

- For PPPoA encapsulation with logical link control (LLC), configure PPP-over-AAL5 LLC encapsulation (`encapsulation atm-ppp-llc`).
- For PPPoA encapsulation with virtual circuit (VC) multiplexing, configure PPP-over-AAL5 multiplex encapsulation (`encapsulation atm-ppp-vc-mux`).

Optionally, you can use dynamic profiles to dynamically or statically apply subscriber services, such as CoS attributes and firewall filters, to the static PPP subscriber interface. Configuring CoS and firewall filters in this manner enables you to efficiently and economically provide these services to PPP subscribers accessing the router over an ATM network.

Concurrent PPP-over-Ethernet-over-ATM and IP-over-Ethernet-over-ATM Configurations

You can configure subscriber interfaces for both PPPoE-over-ATM and IPoE-over-ATM concurrently on a single ATM PVC. IPoE-over-ATM includes support for both IPv4-over-Ethernet-over-ATM interfaces and IPv6-over-Ethernet-over-ATM interfaces.

In concurrent PPPoE-over-ATM and IPoE-over-ATM configurations, you define the ATM logical interface with IPoE-over-ATM encapsulation and specify PPPoE-over-ATM as a supported family. The PPPoE-over-ATM underlying interface with IPoE-over-ATM encapsulation processes PPPoE Discovery packets to establish the PPPoE session. When the PPPoE-over-ATM session is established, the router processes PPPoE-over-ATM session packets and applies PPPoE-over-ATM–specific features on the PPPoE-over-ATM session interface.

To configure concurrent PPPoE-over-ATM and IPoE-over-ATM subscriber interfaces on a single ATM PVC, you configure the ATM logical interface with Ethernet-over-ATM LLC encapsulation (`encapsulation ether-over-atm-llc`). You then configure PPPoE-over-ATM as a supported family. When the router detects the IPoE-over-ATM encapsulation and PPPoE-over-ATM as a supported family, it identifies the configuration as concurrently supporting both PPPoE-over-ATM and IPoE-over-ATM on the same ATM PVC.

The concurrent PPPoE-over-ATM and IPoE-over-ATM configuration supports all features specific to PPPoE-over-ATM interfaces and IPoE-over ATM interfaces, with no changes. These features include the following:

- Class of service (CoS)
- Traffic control profiles with ATM virtual path (VP) shaping and ATM virtual circuit (VC) shaping
- Firewall filters
- PPPoE-over ATM L2TP access concentrator (LAC) support
- Interface statistics
- PPPoE-over-ATM statistics
- Graceful Routing Engine switchover (GRES)
- Unified in-service software upgrade (unified ISSU)
- Dynamic Address Resolution Protocol (ARP)
- Framed IP addresses and address-assignment pools

Configuration and Encapsulation Types for ATM Subscriber Access

You use the same basic statements, commands, and procedures to create, verify, and manage PPPoE-over-ATM, IPoA, IPoE-over-ATM, and PPPoA configurations as the statements, commands, and procedures you use for static configurations on M Series routers and T Series routers, and for dynamic PPPoE configurations on MX Series routers.

A critical element of configuring ATM subscriber access is ensuring that you specify the correct encapsulation type for the ATM logical interface. The encapsulation type you use depends on the supported configuration and, for IPoA and PPPoA configurations, whether you want to configure an encapsulation type that uses logical link control (LLC) or virtual circuit (VC) multiplexing.

ATM Virtual Path Shaping on ATM MICs with SFP

On MX Series routers with Modular Port Concentrator (MPC) interfaces and an ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) installed, you can configure class of service (CoS) hierarchical shaping for the traffic carried on an ATM virtual path (VP). Traffic shaping helps you manage and regulate the traffic flow in your network by shaping the traffic on the VP to a specified rate. With traffic shaping, you can better control the traffic flow to avoid network congestion, and ensure that the traffic adheres to the class-of-service policies you set for it.

To configure hierarchical VP shaping on an ATM MIC with SFP (Model number MIC-3D-8OC3-2OC12-ATM), you must configure an interface set that consists of the ATM logical interface units on the ATM physical interface. The members of the interface set must all share the same virtual path identifier (VPI) and have different virtual circuit identifiers (VCIs). You then define one or more CoS traffic control profiles that include the ATM service category (`atm-service`) and the peak cell rate (`peak-rate`), sustained cell rate (`sustained-rate`), and maximum burst size (`max-burst-size`) parameters.

The ATM service category works in conjunction with the peak cell rate, sustained cell rate, and maximum burst size ATM cell parameters to shape the traffic leaving the interface. Finally, you apply a specified traffic control profile to the output traffic at the interface set and at each of its member ATM logical interfaces.

In the queueing model used for ATM VP hierarchical shaping on ATM MICs with SFP, the ATM physical interface functions as a level 1 scheduler node, the interface set containing the ATM logical interfaces functions as a level 2 scheduler node, and the ATM logical interfaces function as level 3 scheduler nodes.

The following configuration requirements apply to ATM VP shaping on ATM MICs with SFP:

- All ATM interfaces that belong to the same interface set must share the same virtual path identifier (VPI) and have a unique virtual circuit identifier (VCI).
- The ATM interface set can include only ATM interfaces. It cannot include Ethernet interfaces.
- The ATM interface set cannot include PPPoE over ATM interfaces, but it can include the underlying ATM interface over which PPPoE over ATM is carried.

RELATED DOCUMENTATION

ATM for Subscriber Access Encapsulation Types Overview | 427

Understanding Hierarchical Scheduling for MIC and MPC Interfaces

Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM | 439

Example: Configuring a Static PPPoE Subscriber Interface over ATM | 453

Example: Configuring a Static Subscriber Interface for IP Access over ATM | 471

Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM | 480

Example: Configuring a Static PPP Subscriber Interface over ATM | 489

Configuring Concurrent PPPoE-over-ATM and IPoE-over-ATM Subscriber Interfaces on an ATM PVC | 437

Configuring ATM Virtual Path Shaping on ATM MICs with SFP | 432

ATM for Subscriber Access Encapsulation Types Overview

To enable subscriber access to an MX Series router over an ATM network, you can create any of the following configurations on Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use the ATM MIC with SFP:

- PPP-over-Ethernet-over-ATM (PPPoE-over ATM) with a dynamic or static PPPoE (`pp0`) subscriber interface over a static ATM underlying interface
- Routed IP-over-ATM (IPoA) with a static IPv4 or IPv6 subscriber interface over a static ATM underlying interface
- Bridged IP-over-Ethernet-over-ATM (IPoE-over-ATM) with a static IPv4 or IPv6 subscriber interface over a static ATM underlying interface
- PPP-over-ATM (PPPoA) with a static PPP subscriber interface over a static ATM underlying interface
- Concurrent PPP-over-Ethernet-over-ATM interfaces and IP-over-Ethernet-over-ATM interfaces on a single ATM PVC

As part of the configuration procedure, you must specify the appropriate encapsulation type for your configuration on the ATM *logical interface*.

Table 13 on page 427 lists and describes the encapsulation type you must specify as part of the `encapsulation` statement when you configure the ATM logical interface for each supported configuration.

Table 13: Encapsulation Types for Supported ATM Subscriber Access Configurations

ATM Subscriber Access Configuration	Encapsulation Type	Description
PPPoE-over-ATM with dynamic `pp0` subscriber interface	`ppp-over-ether-over-atm-llc`	PPPoE-over-ATM encapsulation with logical link control (LLC)
PPPoE-over-ATM with static `pp0` subscriber interface	`ppp-over-ether-over-atm-llc`	PPPoE-over-ATM encapsulation with LLC
IP-over-ATM (IPoA)	`atm-snap`	ATM subnetwork attachment point (SNAP) encapsulation for IPoA with LLC

Table 13: Encapsulation Types for Supported ATM Subscriber Access Configurations *(Continued)*

ATM Subscriber Access Configuration	Encapsulation Type	Description
	`atm-vc-mux`	ATM VC multiplex encapsulation for IPoA with virtual circuit (VC) multiplexing
IP-over-Ethernet-over-ATM (IPoE-over-ATM) *and* Concurrent IPoE-over-ATM and PPPoE-over-ATM subscriber interfaces on a single ATM VC	`ether-over-atm-llc`	Ethernet-over-ATM encapsulation with LLC
PPP-over-ATM (PPPoA)	`atm-ppp-llc` (for PPPoA with logical link control)	PPP-over-AAL5 encapsulation with LLC
	`atm-ppp-vc-mux` (for PPPoA with virtual circuit multiplexing)	PPP-over-AAL5 encapsulation with VC multiplexing

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421

Configuring ATM for Subscriber Access | 430

Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM | 439

Example: Configuring a Static PPPoE Subscriber Interface over ATM | 453

Example: Configuring a Static Subscriber Interface for IP Access over ATM | 471

Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM | 480

Example: Configuring a Static PPP Subscriber Interface over ATM | 489

Configuring Concurrent PPPoE-over-ATM and IPoE-over-ATM Subscriber Interfaces on an ATM PVC | 437

Guidelines for Configuring ATM for Subscriber Access

The following guidelines apply when you configure PPP-over-Ethernet-over-ATM (PPPoE-over-ATM), IP-over-ATM (IPoA), IP-over-Ethernet-over-ATM (IPoE-over-ATM), PPP-over-ATM (PPPoA), and concurrent PPPoE-over-ATM and IPoE-over-ATM configurations for ATM subscriber access. You can create these configurations on MX Series routers with Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use the ATM MIC with SFP.

For all supported ATM subscriber access configurations:

- Make sure you specify the correct encapsulation type on the ATM *logical interface* for your configuration, as described in "ATM for Subscriber Access Encapsulation Types Overview" on page 427.

For PPPoE-over-ATM configurations:

- For dynamic or static PPPoE-over-ATM configurations, including concurrent PPPoE-over-ATM and IPoE-over-ATM subscriber interfaces on a single ATM PVC, specify PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family pppoe]` hierarchy level. Specifying PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `pppoe-underlying-options]` hierarchy level is not supported for these configurations.

- For dynamic or static PPPoE-over-ATM configurations, you must configure the router to act as a PPPoE server (also known as a *remote access concentrator*). Configuring the router to act as a PPPoE client is not supported in these configurations.

- For dynamic PPPoE-over-ATM configurations, issue the `dynamic-profile` *`profile-name`* statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family pppoe]` hierarchy level to associate the ATM logical interface with the dynamic profile that defines the PPPoE subscriber interface.

For static IPoA and IPoE-over-ATM configurations:

- Specify interface-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet]` hierarchy level (for IPv4) or at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet6]` hierarchy level (for IPv6).

For static PPPoA configurations:

- Specify PPP-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `ppp-options]` hierarchy level.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421

Configuring ATM for Subscriber Access

On MX Series routers with MPC/MIC interfaces that use the ATM MIC with SFP, you can create the following configurations to enable subscribers to access the router over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual connections (PVCs):

- PPP-over-Ethernet-over-ATM (PPPoE-over ATM) with a dynamic PPPoE (`pp0`) subscriber interface over a static ATM underlying interface
- PPP-over-Ethernet-over-ATM (PPPoE-over ATM) with a static PPPoE (`pp0`) subscriber interface over a static ATM underlying interface
- Routed IP-over-ATM (IPoA) with a static IPv4 or IPv6 subscriber interface over a static ATM underlying interface
- Bridged IP-over-Ethernet-over-ATM with a static IPv4 or IPv6 subscriber interface over a static ATM underlying interface
- PPP-over-ATM (PPPoA) with a static PPP subscriber interface over a static ATM underlying interface
- Concurrent PPP-over-Ethernet-over-ATM interfaces and IP-over-Ethernet-over-ATM interfaces on a single ATM PVC

Before you begin:

1. Make sure the MX Series router you are using has Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces and an ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) installed and operational.
    - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.

- For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.

2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.

3. If your configuration includes dynamic profiles for PPPoE, class of service (CoS) attributes, or standard firewall filters, make sure you understand how to configure these attributes and apply them to the subscriber interface.

   - For PPPoE dynamic profiles, see "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194
   - For CoS configuration, see *Configuring Traffic Scheduling and Shaping for Subscriber Access*
   - For standard firewall flter configuration, see Guidelines for Configuring Firewall Filters and Guidelines for Applying Standard Firewall Filters

To configure ATM for subscriber access on an MX Series router:

1. For a PPPoE-over-ATM configuration with a dynamic PPPoE (`pp0`) subscriber interface, create a dynamic profile that defines the `pp0` subscriber interface.

   See "Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM" on page 439.
2. Configure one or more virtual path identifiers (VPIs) on the ATM physical interface.
3. Configure the ATM logical subscriber interface.

   a. Configure the appropriate encapsulation type for your configuration.

      See "ATM for Subscriber Access Encapsulation Types Overview" on page 427.

   b. Configure a virtual circuit identifier (VCI) for each VPI configured on the ATM logical interface.

   c. Configure other interface-specific properties as needed for your configuration.

      See "Guidelines for Configuring ATM for Subscriber Access" on page 429.
4. For static PPPoE-over-ATM configurations, define the static PPPoE (`pp0`) subscriber interface at the `[edit interfaces pp0 unit logical-unit-number]` hierarchy level.

   See "Example: Configuring a Static PPPoE Subscriber Interface over ATM" on page 453.
5. (Optional) Configure RADIUS server options for ATM.

   See *RADIUS Servers and Parameters for Subscriber Access* and *Configuring the RADIUS NAS-Port Extended Format for ATM Interfaces*.
6. (Optional) Verify the configuration for ATM subscriber access.

   See "Verifying and Managing ATM Configurations for Subscriber Access" on page 505.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421

Example: Configuring a Static Subscriber Interface for IP Access over ATM | 471

Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM | 480

Example: Configuring a Static PPP Subscriber Interface over ATM | 489

RADIUS Servers and Parameters for Subscriber Access

Configuring Concurrent PPPoE-over-ATM and IPoE-over-ATM Subscriber Interfaces on an ATM PVC | 437

Configuring ATM Virtual Path Shaping on ATM MICs with SFP

Starting in Junos OS Release 14.2, on MX Series routers with Modular Port Concentrator (MPC) interfaces and an ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) installed, you can configure class-of-service (CoS) hierarchical shaping and schedule for the traffic carried on an ATM virtual path (VP).

After you configure the ATM physical interface and logical interface units, you must configure an interface set that consists of the ATM logical interface units. You then define one or more CoS traffic control profiles that includes the ATM service category (`atm-service`) and the peak cell rate (`peak-rate`), sustained cell rate (`sustained-rate`), and maximum burst size (`max-burst-size`) parameters. Finally, you apply the specified traffic control profile to the output traffic at the interface set and at its member ATM logical interface units.

To configure ATM VP shaping for traffic on an ATM MIC with SFP:

1. Enable CoS hierarchical shaping and scheduling on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port]
user@host# hierarchical-scheduler
```

2. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-fpc/pic/port]
user@host# edit atm-options
```

3. Configure one or more virtual path identifiers (VPIs) on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port atm-options]
user@host# set vpi vpi-identifier
```

4. Configure the appropriate encapsulation type for the ATM logical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set encapsulation encapsulation-type
```

5. Configure one or more virtual circuit identifiers (VCI) for each VPI defined on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set vci vpi-identifier.vci-identifier
```

6. (Optional) Configure PPPoE-specific options as needed for your configuration.

   For example, for PPPoE-over-ATM configurations:

```
[edit interfaces at-fpc/pic/port unit logical-unit-number family pppoe]
user@host# set duplicate-protection
```

> **NOTE**: For dynamic or static PPPoE-over-ATM configurations on MX Series routers, you must specify PPPoE-specific options at the `[edit interfaces interface-name unit logical-unit-number family pppoe]` hierarchy level. Specifying PPPoE-specific options at the `[edit interfaces interface-name unit logical-unit-number pppoe-underlying-options]` hierarchy level is not supported for these configurations.

7. Define the set of ATM logical interfaces for which you want to configure hierarchical schedulers.

   - Specify the name of the ATM interface set.

```
[edit interfaces}
user@host# edit interface-set interface-set-name
```

- Configure each member of the ATM interface set.

```
[edit interfaces interface-set interface-set-name]
user@host# set interface at-fpc/pic/port unit logical-unit-number
```

NOTE: All ATM logical interfaces that belong to the same interface set must share the same VPI and have a unique VCI.

8. Configure one or more traffic shaping and scheduling profiles. For each traffic control profile:

   - Specify the service category that determines the traffic shaping parameter for the ATM queue at the ATM MIC with SFP.

   ```
   [edit class-of-service traffic-control-profiles traffic-control-profile-name]
   user@host# set atm-service (cbr | nrtvbr | rtvbr)
   ```

   - Configure the transmit rate, shaping rate, and default excess rate for the ATM queue.

   ```
   [edit class-of-service traffic-control-profiles traffic-control-profile-name]
   user@host# set peak-rate rate
   user@host# set sustained-rate rate
   user@host# set max-burst-size cells
   ```

   The ATM service category works in conjunction with the `peak-rate`, `sustained-rate`, and `max-burst-size` ATM cell parameters to configure traffic shaping, transmit rate, shaping rate, and default excess rate for an ATM queue.

9. Apply the traffic control profile to the output traffic at the interface set.

   ```
   [edit class-of-service interfaces interface-set interface-set-name]
   user@host# set output-traffic-control-profile profile-name
   ```

10. Apply the traffic control profile to the output traffic at each member interface of the ATM interface set.

   ```
   [edit class-of-service interfaces at-fpc/pic/port unit logical-unit-number]
   user@host# set output-traffic-control-profile profile-name
   ```

The following example configures ATM VP shaping on interface at-1/0/4 with VPI 40. The example defines an ATM interface set named atm-vp-ifset with two member ATM logical interfaces, at-1/0/4.50 and at-1/0/4.51, both of which use VPI 40. Traffic control profiles atm-vp-tcp1, atm-vp-tcp2, and atm-vp-tcp3 are each defined with the `atm-service`, `peak-rate`, `sustained-rate`, and `max-burst size` cell parameters. Finally, the `output-traffic-control-profile` statement applies traffic control profile atm-vp-tcp1 to the output traffic at interface at-1/0/4.50, atm-vp-tcp2 to the output traffic at interface at-1/0/4.51, and atm-vp-tcp3 to the output traffic at the atm-vp-ifset interface set.

```
[edit]
# Configure ATM Physical Interface
user@host# set interfaces at-1/0/4 hierarchical-scheduler
user@host# set interfaces at-1/0/4 atm-options vpi 40
#
# Configure ATM Logical Units
user@host# set interfaces at-1/0/4 unit 50 encapsulation pppoe-over-ether-over-atm-llc
user@host# set interfaces at-1/0/4 unit 50 vci 40.50
user@host# set interfaces at-1/0/4 unit 50 family pppoe duplicate-protection
user@host# set interfaces at-1/0/4 unit 51 encapsulation pppoe-over-ether-over-atm-llc
user@host# set interfaces at-1/0/4 unit 51 vci 40.51
user@host# set interfaces at-1/0/4 unit 51 family pppoe duplicate-protection
#
# Configure ATM Interface Set
user@host# set interfaces interface-set atm-vp-ifset interface at-1/0/4 unit 50
user@host# set interfaces interface-set atm-vp-ifset interface at-1/0/4 unit 51
#
# Configure Traffic Shaping and Scheduling Profiles
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set peak-rate 3k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set sustained-rate 200
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set max-burst-size 1000
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set peak-rate 200
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set sustained-rate 100
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set max-burst-size 150
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set peak-rate 5k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set sustained-rate 1k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set max-burst-size 2000
#
# Apply Traffic Shaping and Scheduling Profiles
user@host# set class-of-service interfaces interface-set atm-vp-ifset output-traffic-control-
profile atm-vp-tcp3
```

```
user@host# set class-of-service interfaces at-1/0/4 unit 50 output-traffic-control-profile atm-
vp-tcp1
user@host# set class-of-service interfaces at-1/0/4 unit 51 output-traffic-control-profile atm-
vp-tcp2
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting in Junos OS Release 14.2, on MX Series routers with Modular Port Concentrator (MPC) interfaces and an ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) installed, you can configure class-of-service (CoS) hierarchical shaping and schedule for the traffic carried on an ATM virtual path (VP).

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421

Configuring CoS on Circuit Emulation ATM MICs

CoS on Circuit Emulation ATM MICs Overview

CHAPTER 33

Configuring PPPoE Subscriber Interfaces Over ATM

IN THIS CHAPTER

Configuring Concurrent PPPoE-over-ATM and IPoE-over-ATM Subscriber Interfaces on an ATM PVC

To configure concurrent PPPoE-over-ATM and IPoE-over-ATM subscriber interfaces on a single ATM PVC, you configure the ATM logical interface as an IPoE-over-ATM interface by specifying the `ether-over-atm-llc` encapsulation type. You then use the `family pppoe` stanza at the `[edit interfaces at-`*`fpc/pic/port`*` unit `*`logical-unit-number`*`]` hierarchy level to configure PPPoE-over-ATM as a supported family.

When the router detects the `family pppoe` stanza and the IPoE-over-ATM encapsulation, it identifies the configuration as concurrently supporting both PPPoE-over-ATM and IPoE-over-ATM on the same ATM PVC.

Before you begin:

Configure a PPPoE dynamic profile.

See "Configuring a PPPoE Dynamic Profile" on page 195.

To configure concurrent PPPoE-over-ATM and IPoE-over-ATM subscriber interfaces on an ATM PVC:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-fpc/pic/port]
user@host# edit atm-options
```

2. Configure one or more VPIs on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port atm-options]
user@host# set vpi vpi-identifier
```

3. Configure IPoE-over-ATM encapsulation on the ATM logical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set encapsulation ether-over-atm-llc
```

4. Configure the VCI for the ATM logical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set vci vpi-identifier.vci-identifier
```

5. Configure one or both of the following IP protocol families and addresses as appropriate for your network configuration.

   - For IPv4 (`inet`):

     ```
     [edit interfaces at-fpc/pic/port unit logical-unit-number]
     user@host# set family inet address address
     ```

   - For IPv6 (`inet6`):

     ```
     [edit interfaces at-fpc/pic/port unit logical-unit-number]
     user@host# set family inet6 address address
     ```

6. Configure PPPoE-over-ATM as a supported family by associating a PPPoE dynamic profile with the ATM logical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set family pppoe dynamic-profile profile-name
```

The dynamic profile defines PPPoE-specific options for the `pp0` logical interface, and establishes the PPPoE session. When the PPPoE-over-ATM session is established, PPPoE-over-ATM features operate on the PPPoE-over-ATM session interface.

7. Enable the IPv6 neighbor discovery protocol for the ATM logical interface.

```
[edit protocols router-advertisement interface at-fpc/pic/port.logical-unit-number]
user@host# set prefix prefix
```

The following example configures concurrent support for IPv4-over-Ethernet-over-ATM, IPv6-over-Ethernet-over-ATM, and PPPoE-over-ATM subscriber interfaces on an ATM PVC with VPI 10 and VCI 200. ATM logical interface at-1/2/0.200 is configured with IPoE-over-ATM encapsulation (`ether-over-atm-llc`). The `family pppoe` statement configures PPPoE-over-ATM as a supported family by associating a PPPoE dynamic profile named pppoeoa-profile with interface at-1/2/0.200.

```
[edit]
user@host# set interfaces at-1/2/0 atm-options vpi 10
user@host# set interfaces at-1/2/0 unit 200 encapsulation ether-over-atm-llc
user@host# set interfaces at-1/2/0 unit 200 vci 10.200
user@host# set interfaces at-1/2/0 unit 200 family inet address 10.101.103.1/24
user@host# set interfaces at-1/2/0 unit 200 family inet6 address 201.db8:13:13::1/64
user@host# set interfaces at-1/2/0 unit 200 family pppoe dynamic-profile pppoeoa-profile
user@host# set protocols router-advertisement interface at-1/2/0.200 prefix 201.db8:13:13::/64
```

RELATED DOCUMENTATION

Guidelines for Configuring ATM for Subscriber Access | 429

Verifying and Managing ATM Configurations for Subscriber Access | 505

ATM for Subscriber Access Overview | 421

Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM

IN THIS SECTION

This example illustrates a Point-to-Point Protocol over Ethernet (PPPoE) over ATM configuration that creates a dynamic PPPoE (`pp0`) subscriber interface over a static ATM underlying interface on an MX Series router. The router must have Module Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use an ATM MIC with small form-factor pluggable transceiver (SFP).

NOTE: You can also configure a *static* PPPoE interface over a static ATM underlying interface on an MX Series router with an ATM MIC with SFP installed. For information, see "Example: Configuring a Static PPPoE Subscriber Interface over ATM" on page 453.

Requirements

This example uses the following software and hardware components:

- MX Series 5G Universal Routing Platform
- ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) and compatible MPC1 or MPC2

Before you begin:

1. Make sure the MX Series router you are using has an ATM MIC with SFP installed and operational.
   - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.
   - For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.
2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.
3. Make sure you understand how to configure and use dynamic PPPoE subscriber interfaces.
   - For overview information, see "Subscriber Interfaces and PPPoE Overview" on page 186
   - For configuration instructions, see "Configuring Dynamic PPPoE Subscriber Interfaces" on page 194

Overview

By using the ATM MIC with SFP and a supported MPC, you can configure an MX Series router to support dynamic PPPoE subscriber access over an ATM network. PPPoE-over-ATM configurations on MX Series routers consist of one or more dynamically created PPPoE (`pp0`) subscriber interfaces over a static ATM underlying interface. Most PPPoE and subscriber services features supported on terminated connections and tunneled (L2TP access concentrator, or LAC) connections are also supported for PPPoE-over-ATM connections on an MX Series router.

Optionally, you can dynamically apply subscriber services such as class of service (CoS) and firewall filters to the PPPoE subscriber interface by configuring these services in the dynamic profile that creates the `pp0` subscriber interface. In this example, the PPPoE dynamic profile (pppoe-profile) applies CoS traffic shaping parameters to the dynamic `pp0` subscriber interface. Configuring CoS and firewall filters in this manner enables you to efficiently and economically provide these services to PPPoE subscribers accessing the router over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual connections (PVCs).

This example includes the following basic steps to configure dynamic PPPoE-over-ATM subscriber access on an MX Series router:

1. Create a PPPoE dynamic profile named pppoe-profile for the `pp0` subscriber interface that includes all of the following:

   - The logical unit number, represented by the `$junos-interface-unit` predefined dynamic variable

   - The name of the underlying ATM interface, represented by the `$junos-underlying-interface` predefined dynamic variable

   - The `server` statement, which configures the router to act as a PPPoE server

     NOTE: Configuring the router to act as a PPPoE client is not supported.

   - The unnumbered address (lo0.0) for the IPv4 (`inet`) protocol family

   - CoS traffic shaping parameters

2. Statically configure the ATM physical interface at-1/0/0 with virtual path identifier (VPI) 3.

3. Statically configure logical unit 2 on the ATM physical interface (at-1/0/0.2) with at least the following properties:

   - PPPoE-over-ATM logical link control (LLC) encapsulation (`ppp-over-ether-over-atm-llc`)

   - Virtual circuit identifier (VCI) 2 on VPI 3. The combination of VPIs and VCIs provisions the ATM AAL5 PVC for access over the ATM network.

   - PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family pppoe]` hierarchy level, including at least the name of the associated PPPoE dynamic profile (pppoe-profile) that creates the pp0 dynamic subscriber interface

     In dynamic PPPoE-over-ATM configurations, each `pp0` interface defined in the dynamic profile corresponds to a dynamic PPPoE subscriber interface.

NOTE: For dynamic or static PPPoE-over-ATM configurations on MX Series routers, You must specify PPPoE-specific options in the `family pppoe` stanza at the `[edit interfaces *interface-name* unit *logical-unit-number*]` hierarchy level. Specifying PPPoE-specific options in the `pppoe-underlying-options` stanza at the `[edit interfaces *interface-name* unit *logical-unit-number*]` hierarchy level is not supported for these configurations.

Configuration

IN THIS SECTION

To configure a dynamic PPPoE subscriber interface over an underlying ATM interface, perform these tasks:

CLI Quick Configuration

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

```
# PPPoE Dynamic Profile
set dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit" ppp-options chap
set dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit" pppoe-options
underlying-interface "$junos-underlying-interface"
set dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit" pppoe-options
server
set dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit" no-keepalives
set dynamic-profiles pppoe-profile interfaces pp0 unit "$junos-interface-unit" family inet
unnumbered-address lo0.0
set dynamic-profiles pppoe-profile class-of-service traffic-control-profiles tcp-test shaping-
rate 10m
set dynamic-profiles pppoe-profile class-of-service interfaces pp0 unit "$junos-interface-unit"
```

```
output-traffic-control-profile tcp-test
#
# ATM Physical Interface
set interfaces at-1/0/0 atm-options vpi 3
#
# Logical Unit 2
set interfaces at-1/0/0 atm-options vpi 3
set interfaces at-1/0/0 unit 2 encapsulation ppp-over-ether-over-atm-llc
set interfaces at-1/0/0 unit 2 vci 3.2
set interfaces at-1/0/0 unit 2 family pppoe access-concentrator ac-pppoeoa
set interfaces at-1/0/0 unit 2 family pppoe duplicate-protection
set interfaces at-1/0/0 unit 2 family pppoe dynamic-profile pppoe-profile
set interfaces at-1/0/0 unit 2 family pppoe max-sessions 3
set interfaces at-1/0/0 unit 2 family pppoe short-cycle-protection
```

Configuring the PPPoE Dynamic Profile

Step-by-Step Procedure

To configure the PPPoE dynamic profile for the pp0 subscriber interface:

1. Name the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles pppoe-profile
```

2. Specify that you want to configure the pp0 (PPPoE) interface.

```
[edit dynamic-profiles pppoe-profile]
user@host# edit interfaces pp0
```

3. Specify that you want to configure the logical unit represented by the $junos-interface-unit predefined variable.

```
[edit dynamic-profiles pppoe-profile interfaces pp0]
user@host# edit unit $junos-interface-unit
```

The $junos-interface-unit variable is dynamically replaced with the actual unit number supplied by the network when the subscriber logs in.

4. Configure PPPoE-specific options for the `pp0` interface.

   a. Configure the ATM underlying interface represented by the `$junos-underlying-interface` predefined variable.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit”]
   user@host# set pppoe-options underlying-interface $junos-underlying-interface
   ```

   The `$junos-underlying-interface` variable is dynamically replaced with the actual name of the underlying interface supplied by the network when the subscriber logs in.

   b. Configure the router to act as a PPPoE server, also known as a remote access concentrator.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit”]
   user@host# set pppoe-options server
   ```

5. Configure Challenge Handshake Authentication Protocol (CHAP) authentication for the `pp0` interface.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit”]
   user@host# set ppp-options chap
   ```

6. Disable sending keepalive messages on the interface.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit”]
   user@host# set no-keepalives
   ```

7. Configure the protocol family for the `pp0` interface.

   a. Specify that you want to configure the IPv4 (`inet`) protocol family.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit”]
   user@host# edit family inet
   ```

   b. Configure the unnumbered address for the protocol family.

   ```
   [edit dynamic-profiles pppoe-profile interfaces pp0 unit “$junos-interface-unit” family
   inet]
   ```

```
user@host# set unnumbered-address lo0.0
user@host# up 4
```

8. Configure CoS traffic shaping parameters in the dynamic profile for the pp0 subscriber interface.

   a. Specify that you want to configure CoS traffic shaping parameters.

   ```
   [edit dynamic-profiles pppoe-profile]
   user@host# edit class-of-service
   ```

   b. Create a traffic-control profile.

   ```
   [edit dynamic-profiles pppoe-profile class-of-service]
   user@host# edit traffic-control-profiles tcp-test
   ```

   c. Configure the traffic shaping rate.

   ```
   [edit dynamic-profiles pppoe-profile class-of-service traffic-control-profiles tcp-test]
   user@host# set shaping-rate 10m
   user@host# up 2
   ```

   d. Apply the traffic shaping parameters to the pp0 dynamic subscriber interface.

   ```
   [edit dynamic-profiles pppoe-profile class-of-service]
   user@host# edit interfaces pp0 unit $junos-interface-unit
   ```

   e. Apply the output traffic scheduling and shaping profile to the interface.

   ```
   [edit dynamic-profiles pppoe-profile class-of-service interfaces pp0 unit “$junos-
   interface-unit]
   user@host# set output-traffic-control-profile tcp-test
   ```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the PPPoE dynamic profile configuration by issuing the `show dynamic-profiles pppoe-profile` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show dynamic-profiles pppoe-profile
interfaces {
    pp0 {
        unit "$junos-interface-unit" {
            ppp-options {
                chap;
            }
            pppoe-options {
                underlying-interface "$junos-underlying-interface";
                server;
            }
            no-keepalives;
            family inet {
                unnumbered-address lo0.0;
            }
        }
    }
}
class-of-service {
    traffic-control-profiles {
        tcp-test {
            shaping-rate 10m;
        }
    }
    interfaces {
        pp0 {
            unit "$junos-interface-unit" {
                output-traffic-control-profile tcp-test;
            }
        }
    }
}
```

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

Configuring the ATM Physical Interface

Step-by-Step Procedure

To configure the ATM physical interface:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-1/0/0]
user@host# edit atm-options
```

2. Configure one or more VPIs on the physical interface.

```
[edit interfaces at-1/0/0 atm-options]
user@host# set vpi 3
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the ATM physical interface configuration by issuing the `show interfaces at-1/0/0` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/0
atm-options {
    vpi 3;
}
```

If you are done configuring the ATM physical interface, enter `commit` from configuration mode.

Configuring the Dynamic PPPoE Subscriber Interface on Logical Unit 2

Step-by-Step Procedure

To configure the dynamic PPPoE subscriber interface on logical unit 2:

1. Configure PPPoE-over-ATM LLC encapsulation on the interface.

```
[edit interfaces at-1/0/0 unit 2]
user@host# set encapsulation ppp-over-ether-over-atm-llc
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/0 unit 2]
user@host# set vci 3.2
```

   This statement configures VCI 2 on VPI 3.

3. Specify that you want to configure the PPPoE protocol family.

```
[edit interfaces at-1/0/0 unit 2]
user@host# edit family pppoe
```

4. Associate the interface with the dynamic profile that creates the dynamic PPPoE subscriber interface.

```
[edit interfaces at-1/0/0 unit 2 family pppoe]
user@host# set dynamic-profile pppoe-profile
```

5. Configure additional PPPoE-specific options for the dynamic subscriber interface.

```
[edit interfaces at-1/0/0 unit 2 family pppoe]
user@host# set max-sessions 3
user@host# set duplicate-protection
user@host# set short-cycle-protection
user@host# set access-concentrator ac-pppoeoa
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the dynamic PPPoE subscriber interface configuration on logical unit 2 by issuing the `show interfaces at-1/0/0.2` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/0.2
encapsulation ppp-over-ether-over-atm-llc;
vci 3.2;
family pppoe {
    access-concentrator ac-pppoeoa;
    duplicate-protection;
    dynamic-profile pppoe-profile;
    max-sessions 3;
    short-cycle-protection;
}
```

If you are done configuring the dynamic PPPoE subscriber interface on logical unit 2, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the dynamic PPPoE subscriber interface is properly configured on ATM interface at-1/0/0.2, perform the following tasks:

Verifying the ATM Physical Interface Configuration

Purpose

Verify that ATM physical interface at-1/0/0 is properly configured for use with ATM PVCs.

Action

From operational mode, issue the `show interfaces at-1/0/0` command.

For brevity, this `show` command output includes only the configuration that is relevant to the at-1/0/0 physical interface. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show interfaces at-1/0/0
Physical interface: at-1/0/0, Enabled, Physical link is Up
  Interface index: 173, SNMP ifIndex: 592
  Link-level type: ATM-PVC, MTU: 2048, Clocking: Internal, SDH mode, Speed: OC3, Loopback: None,
Payload scrambler: Enabled
  Device flags   : Present Running
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 00:00:5e:00:53:95
  Last flapped   : 2012-09-17 07:21:19 PDT (08:26:16 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SDH   alarms   : None
  SDH   defects  : None
    VPI 3
      Flags: Active
      Total down time: 0 sec, Last down: Never
  Traffic statistics:
         Input  packets:                    0
         Output packets:                    0
...
```

Meaning

`ATM-PVC` in the Link-level Type field indicates that encapsulation for ATM permanent virtual circuits is being used on ATM physical interface at-1/0/0. The `Active` flag for VPI 3 indicates that the virtual path is up and operational.

Verifying the Dynamic PPPoE Subscriber Interface Configuration on Logical Unit 2

Purpose

Verify that the dynamic PPPoE subscriber interface is properly configured on logical unit 2 (at-1/0/0.2).

Action

From operational mode, issue the `show interfaces at-1/0/0.2` command.

```
user@host> show interfaces at-1/0/0.2
  Logical interface at-1/0/0.2 (Index 350) (SNMP ifIndex 1701)
    Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE-over-ATM-LLC
    Input packets : 0
    Output packets: 0
    Protocol pppoe
      Dynamic Profile: pppoe-profile,
      Service Name Table: None,
      Max Sessions: 3, Max Sessions VSA Ignore: Off,
      Duplicate Protection: On, Short Cycle Protection: mac-address,
      AC Name: ac-pppoeoa
    VCI 3.2
      Flags: Active
      Total down time: 0 sec, Last down: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`PPPoE-over-ATM-LLC` in the Encapsulation field indicates that logical interface at-1/0/0.2 is properly configured for PPPoE-over-ATM LLC encapsulation. `Protocol pppoe` indicates that the PPPoE protocol family has been properly configured on the logical interface. The Dynamic Profile field indicates that dynamic profile `pppoe-profile` creates the dynamic PPPoE subscriber interface. The `Active` flag for VCI 3.2 indicates that VCI 2 on VPI 3 is up and operational.

Verifying the PPPoE Underlying Interface Configuration

Purpose

Verify that the underlying interface is properly configured for dynamic PPPoE-over-ATM subscriber access.

Action

From operational mode, issue the `show pppoe underlying-interfaces at-1/0/0.2` command.

```
user@host> show pppoe underlying-interfaces at-1/0/0.2 detail
at-1/0/0.2 Index 350
  State: Static, Dynamic Profile: pppoe-profile,
  Max Sessions: 3, Max Sessions VSA Ignore: Off,
  Active Sessions: 0,
  Service Name Table: None,
  Duplicate Protection: On, Short Cycle Protection: mac-address,
  AC Name: ac-pppoeoa,
```

Meaning

This command indicates that ATM logical interface at-1/0/0.2 is properly configured as the PPPoE underlying interface. `Static` in the State field indicates that at-1/0/0/.2 is statically configured. The Dynamic Profile field indicates that `pppoe-profile` is the name of the dynamic profile used to create this interface. The remaining fields display information about the PPPoE-specific interface options configured for the PPPoE underlying interface at the `[edit interfaces at-1/0/0 unit 2 family pppoe]` hierarchy level.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421
Configuring ATM for Subscriber Access | 430
Example: Configuring a Static PPPoE Subscriber Interface over ATM | 453
Example: Configuring a Static Subscriber Interface for IP Access over ATM | 471
Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM | 480
Example: Configuring a Static PPP Subscriber Interface over ATM | 489

Example: Configuring a Static PPPoE Subscriber Interface over ATM

IN THIS SECTION

This example illustrates a Point-to-Point Protocol over Ethernet (PPPoE) over ATM configuration that creates a static PPPoE (`pp0`) subscriber interface over a static ATM underlying interface on an MX Series router. The router must have Module Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use an ATM MIC with small form-factor pluggable transceiver (SFP).

NOTE: You can also configure a *dynamic* PPPoE interface over a static ATM underlying interface on an MX Series router with an ATM MIC with SFP installed. For information, see "Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM" on page 439.

Requirements

This example uses the following software and hardware components:

- MX Series 5G Universal Routing Platform
- ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) and compatible MPC1 or MPC2

Before you begin:

1. Make sure the MX Series router you are using has an ATM MIC with SFP installed and operational.
   - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.
   - For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.
2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.

Overview

By using the ATM MIC with SFP and a supported MPC, you can configure an MX Series router to support static PPPoE subscriber access over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual connections (PVCs). PPPoE-over-ATM configurations on MX Series routers consist of one or more statically created PPPoE (`pp0`) logical subscriber interfaces over a static ATM underlying interface. Most PPPoE and subscriber services features supported on terminated connections and tunneled (L2TP access concentrator, or LAC) connections are also supported for PPPoE-over-ATM connections on an MX Series router.

This example include the following basic steps to configure static PPPoE-over-ATM subscriber access on an MX Series router:

1. Statically configure ATM physical interface at-1/0/6 with virtual path identifier (VPI) 6.

2. Statically configure logical unit 2 on the ATM physical interface (at-1/0/6.2) with the following properties:

   - PPPoE-over-ATM logical link control (LLC) encapsulation (`ppp-over-ether-over-atm-llc`)

   - Virtual circuit identifier (VCI) 2 on VPI 6. The combination of VPIs and VCIs provisions the ATM AAL5 PVC for access over the ATM network.

   - (Optional) PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family pppoe]` hierarchy level

   > **NOTE**: For dynamic or static PPPoE-over-ATM configurations on MX Series routers, You must specify PPPoE-specific options in the `family pppoe` stanza at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level. Specifying PPPoE-specific options in the `pppoe-underlying-options` stanza at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level is not supported for these configurations.

3. Statically configure the `pp0` logical subscriber interface (pp0.2) with at least the following properties:

   - The name of the underlying ATM interface (at-1/0/6.2)

   - The `server` statement, which configures the router to act as a PPPoE server

   - The unnumbered address (lo0.0) for the `inet` (IPv4) or `inet6` (IPv6) protocol family

   In static PPPoE-over-ATM configurations, each `pp0` logical interface configured at the `[edit interfaces pp0 unit` *`logical-unit-number`*`]` hierarchy level corresponds to a static PPPoE subscriber interface.

Configuration

IN THIS SECTION

To configure a static PPPoE subscriber interface over an underlying ATM interface, perform these tasks:

CLI Quick Configuration

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

```
# ATM Physical Interface
set interfaces at-1/0/6 atm-options vpi 6
#
# Logical Unit 2
set interfaces at-1/0/6 unit 2 encapsulation ppp-over-ether-over-atm-llc
set interfaces at-1/0/6 unit 2 vci 6.2
set interfaces at-1/0/6 unit 2 family pppoe access-concentrator ac-pppoeoa
set interfaces at-1/0/6 unit 2 family pppoe duplicate-protection
set interfaces at-1/0/6 unit 2 family pppoe max-sessions 3
set interfaces at-1/0/6 unit 2 family pppoe max-sessions-vsa-ignore
set interfaces at-1/0/6 unit 2 family pppoe short-cycle-protection lockout-time-min 120
set interfaces at-1/0/6 unit 2 family pppoe short-cycle-protection lockout-time-max 240
#
# Static PPPoE Subscriber Interface
set interfaces pp0 unit 2 ppp-options chap
set interfaces pp0 unit 2 pppoe-options underlying-interface at-1/0/6.2
set interfaces pp0 unit 2 pppoe-options server
set interfaces pp0 unit 2 keepalives interval 10
set interfaces pp0 unit 2 family inet unnumbered-address lo0.0
```

Configuring the ATM Physical Interface

Step-by-Step Procedure

To configure the ATM physical interface:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-1/0/6]
user@host# edit atm-options
```

2. Configure one or more VPIs on the physical interface.

```
[edit interfaces at-1/0/6 atm-options]
user@host# set vpi 6
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the ATM physical interface configuration by issuing the `show interfaces at-1/0/6` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/6
atm-options {
    vpi 6;
}
```

If you are done configuring the ATM physical interface, enter `commit` from configuration mode.

Configuring Encapsulation, VCI, and PPPoE Options on Logical Unit 2

Step-by-Step Procedure

To configure encapsulation, VCI, and PPPoE options on logical unit 2:

1. Configure PPPoE-over-ATM LLC encapsulation on the interface.

```
[edit interfaces at-1/0/6 unit 2]
user@host# set encapsulation ppp-over-ether-over-atm-llc
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/6 unit 2]
user@host# set vci 6.2
```

   This statement configures VCI 2 on VPI 6.

3. Specify that you want to configure the PPPoE protocol family.

```
[edit interfaces at-1/0/6 unit 2]
user@host# edit family pppoe
```

4. Configure additional PPPoE-specific options for the dynamic subscriber interface.

```
[edit interfaces at-1/0/6 unit 2 family pppoe]
user@host# set duplicate-protection
user@host# set short-cycle-protection lockout-time-min 120 lockout-time-max 240
user@host# set max-sessions 3
user@host# set max-sessions-vsa-ignore
user@host# set access-concentrator ac-pppoeoa
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the configuration on logical unit 2 by issuing the `show interfaces at-1/0/6.2` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/6.2
encapsulation ppp-over-ether-over-atm-llc;
vci 6.2;
family pppoe {
    access-concentrator ac-pppoeoa;
```

```
    duplicate-protection;
    max-sessions 3;
    max-sessions-vsa-ignore;
    short-cycle-protection {
        lockout-time-min 120;
        lockout-time-max 240;
    }
}
```

If you are done configuring logical unit 2, enter `commit` from configuration mode.

Configuring the Static PPPoE Subscriber Interface

Step-by-Step Procedure

To configure the static PPPoE subscriber interface:

1. Specify that you want to configure the `pp0` subscriber interface on logical unit 2.

   ```
   [edit]
   user@host# edit interfaces pp0 unit 2
   ```

2. Specify that you want to configure PPP options for the subscriber interface.

   ```
   [edit interfaces pp0 unit 2]
   user@host# edit ppp-options
   ```

3. Configure Challenge Handshake Authentication Protocol (CHAP) authentication for the subscriber interface.

   ```
   [edit interfaces pp0 unit 2 ppp-options]
   user@host# set chap
   user@host# up
   ```

4. Specify that you want to configure PPPoE-specific options.

   ```
   [edit interfaces pp0 unit 2]
   user@host# edit pppoe-options
   ```

5. Associate the PPPoE subscriber interface with the underlying ATM interface.

```
[edit interfaces pp0 unit 2 pppoe-options]
user@host# set underlying-interface at-1/0/6.2
```

6. Configure the router to act as a PPPoE server, also known as a remote access concentrator.

```
[edit interfaces pp0 unit 2 pppoe-options]
user@host# set server
user@host# up
```

7. Configure the interval for sending keepalive requests.

```
[edit interfaces pp0 unit 2]
user@host# set keepalives interval 10
```

8. Specify that you want to configure the IPv4 (`inet`) protocol family.

```
[edit interfaces pp0 unit 2]
user@host# edit family inet
```

9. Configure the unnumbered address for the protocol family.

```
[edit interfaces pp0 unit 2 family inet]
user@host# set unnumbered-address lo0.0
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static PPPoE subscriber interface configuration by issuing the `show interfaces pp0` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces pp0
unit 2 {
    ppp-options {
```

```
            chap;
        }
        pppoe-options {
            underlying-interface at-1/0/6.2;
            server;
        }
        keepalives interval 10;
        family inet {
            unnumbered-address lo0.0;
        }
    }
```

If you are done configuring the static PPPoE subscriber interface, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the static PPPoE subscriber interface pp0.2 is properly configured on ATM underlying interface at-1/0/6.2, perform the following tasks:

Verifying the ATM Physical Interface Configuration

Purpose

Verify that ATM physical interface at-1/0/6 is properly configured for use with ATM PVCs.

Action

From operational mode, issue the `show interfaces at-1/0/6` command.

For brevity, this show command output includes only the configuration that is relevant to the at-1/0/6 physical interface. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show interfaces at-1/0/6
Physical interface: at-1/0/6, Enabled, Physical link is Down
  Interface index: 179, SNMP ifIndex: 598
  Link-level type: ATM-PVC, MTU: 2048, Clocking: Internal, SDH mode, Speed: OC3, Loopback: None,
Payload scrambler: Enabled
  Device flags   : Present Running Down
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 00:00:5e:00:53:9b
  Last flapped   : 2012-09-19 07:57:59 PDT (07:46:56 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SDH   alarms   : LOL, LOS
  SDH   defects  : LOL, LOS, LOP, BERR-SF, HP-FERF
    VPI 6
      Flags: Active
      Total down time: 0 sec, Last down: Never
  Traffic statistics:
         Input  packets:                    0
         Output packets:                    0
...
```

Meaning

ATM-PVC in the Link-level Type field indicates that encapsulation for ATM permanent virtual circuits is being used on ATM physical interface at-1/0/6. The Active flag for VPI 6 indicates that the virtual path is up and operational.

Verifying the Encapsulation, VCI, and PPPoE Options Configuration on Logical Unit 2

Purpose

Verify that the encapsulation, VCI, and PPPoE settings have been properly configured on logical unit 2 (at-1/0/6.2).

Action

From operational mode, issue the `show interfaces at-1/0/6.2` command.

```
user@host> show interfaces at-1/0/6.2
  Logical interface at-1/0/6.2 (Index 345) (SNMP ifIndex 1990)
    Flags: Device-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE-over-ATM-LLC
    Input packets : 0
    Output packets: 0
    Protocol pppoe
      Dynamic Profile: None,
      Service Name Table: None,
      Max Sessions: 3, Max Sessions VSA Ignore: On,
      Duplicate Protection: On, Short Cycle Protection: mac-address,
      AC Name: ac-pppoeoa
    VCI 6.2
      Flags: Active
      Total down time: 0 sec, Last down: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`PPPoE-over-ATM-LLC` in the Encapsulation field indicates that logical interface at-1/0/6.2 is properly configured for PPPoE-over-ATM LLC encapsulation. `Protocol pppoe` indicates that the PPPoE protocol family has been properly configured on the logical interface. The `Active` flag for VCI 6.2 indicates that VCI 2 on VPI 6 is up and operational.

Verifying the Static PPPoE Subscriber Interface Configuration

Purpose

Verify that the static PPPoE subscriber interface (pp0.2) is properly configured.

Action

From operational mode, issue the `show interfaces pp0` command.

```
user@host> show interfaces pp0
Physical interface: pp0, Enabled, Physical link is Up
```

```
  Interface index: 131, SNMP ifIndex: 505
  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

  Logical interface pp0.2 (Index 360) (SNMP ifIndex 1991)
    Flags: Hardware-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPPoE
    PPPoE:
      State: SessionDown, Session ID: None,
      Underlying interface: at-1/0/6.2 (Index 345)
    Input packets : 0
    Output packets: 0
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  LCP state: Not-configured
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured, mpls: Not-
configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 1492
      Flags: Sendbcast-pkt-to-re, Protocol-Down
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 198.51.100/24, Local: 198.51.100.11
```

Meaning

PPPoE in the Link-level type field indicates that PPPoE encapsulation is in use on the pp0 physical interface. PPPoE in the Encapsulation field indicates that PPPoE encapsulation is also in use on the pp0.2 logical subscriber interface. The Underlying interface field indicates that at-1/0/6.2 is properly configured as the underlying interface for the static PPPoE subscriber interface. Protocol inet indicates that the IPv4 protocol family is properly configured on the pp0.2 logical subscriber interface.

Verifying the PPPoE Underlying Interface Configuration

Purpose

Verify that the underlying interface is properly configured for static PPPoE-over-ATM subscriber access.

Action

From operational mode, issue the `show pppoe underlying-interfaces at-1/0/6.2 extensive` command.

```
user@host> show pppoe underlying-interfaces at-1/0/6.2 extensive
at-1/0/6.2 Index 345
  State: Static, Dynamic Profile: None,
  Max Sessions: 3, Max Sessions VSA Ignore: On,
  Active Sessions: 0,
  Service Name Table: None,
  Duplicate Protection: On, Short Cycle Protection: mac-address,
  AC Name: ac-pppoeoa,
  PacketType                       Sent         Received
    PADI                              0                0
    PADO                              0                0
    PADR                              0                0
    PADS                              0                0
    PADT                              0                0
    Service name error                0                0
    AC system error                   0                0
    Generic error                     0                0
    Malformed packets                 0                0
    Unknown packets                   0                0
  Lockout Time (sec):  Min: 120, Max: 240
    Total clients in lockout: 0
    Total clients in lockout grace period: 0
```

Meaning

This command indicates that ATM logical interface at-1/0/6.2 is properly configured as the PPPoE underlying interface. `Static` in the State field indicates that at-1/0/0/.2 is statically configured. The remaining fields display information about the PPPoE-specific interface options configured for the PPPoE underlying interface at the `[edit interfaces at-1/0/6 unit 2 family pppoe]` hierarchy level. The Lockout Time fields, which appear in this command only when you display the `extensive` level of output, show the minimum lockout time (120 seconds) and maximum lockout time (240 seconds) configured for the PPPoE underlying interface.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421

CHAPTER 34

Configuring ATM Virtual Path Shaping on ATM MICs with SFP

IN THIS CHAPTER

Configuring ATM Virtual Path Shaping on ATM MICs with SFP

Starting in Junos OS Release 14.2, on MX Series routers with Modular Port Concentrator (MPC) interfaces and an ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) installed, you can configure class-of-service (CoS) hierarchical shaping and schedule for the traffic carried on an ATM virtual path (VP).

After you configure the ATM physical interface and logical interface units, you must configure an interface set that consists of the ATM logical interface units. You then define one or more CoS traffic control profiles that includes the ATM service category (`atm-service`) and the peak cell rate (`peak-rate`), sustained cell rate (`sustained-rate`), and maximum burst size (`max-burst-size`) parameters. Finally, you apply the specified traffic control profile to the output traffic at the interface set and at its member ATM logical interface units.

To configure ATM VP shaping for traffic on an ATM MIC with SFP:

1. Enable CoS hierarchical shaping and scheduling on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port]
user@host# hierarchical-scheduler
```

2. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-fpc/pic/port]
user@host# edit atm-options
```

3. Configure one or more virtual path identifiers (VPIs) on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port atm-options]
user@host# set vpi vpi-identifier
```

4. Configure the appropriate encapsulation type for the ATM logical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set encapsulation encapsulation-type
```

5. Configure one or more virtual circuit identifiers (VCI) for each VPI defined on the ATM physical interface.

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
user@host# set vci vpi-identifier.vci-identifier
```

6. (Optional) Configure PPPoE-specific options as needed for your configuration.

   For example, for PPPoE-over-ATM configurations:

```
[edit interfaces at-fpc/pic/port unit logical-unit-number family pppoe]
user@host# set duplicate-protection
```

> **NOTE**: For dynamic or static PPPoE-over-ATM configurations on MX Series routers, you must specify PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family pppoe]` hierarchy level. Specifying PPPoE-specific options at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `pppoe-underlying-options]` hierarchy level is not supported for these configurations.

7. Define the set of ATM logical interfaces for which you want to configure hierarchical schedulers.

   - Specify the name of the ATM interface set.

```
[edit interfaces}
user@host# edit interface-set interface-set-name
```

- Configure each member of the ATM interface set.

```
[edit interfaces interface-set interface-set-name]
user@host# set interface at-fpc/pic/port unit logical-unit-number
```

NOTE: All ATM logical interfaces that belong to the same interface set must share the same VPI and have a unique VCI.

8. Configure one or more traffic shaping and scheduling profiles. For each traffic control profile:

   - Specify the service category that determines the traffic shaping parameter for the ATM queue at the ATM MIC with SFP.

   ```
   [edit class-of-service traffic-control-profiles traffic-control-profile-name]
   user@host# set atm-service (cbr | nrtvbr | rtvbr)
   ```

   - Configure the transmit rate, shaping rate, and default excess rate for the ATM queue.

   ```
   [edit class-of-service traffic-control-profiles traffic-control-profile-name]
   user@host# set peak-rate rate
   user@host# set sustained-rate rate
   user@host# set max-burst-size cells
   ```

   The ATM service category works in conjunction with the `peak-rate`, `sustained-rate`, and `max-burst-size` ATM cell parameters to configure traffic shaping, transmit rate, shaping rate, and default excess rate for an ATM queue.

9. Apply the traffic control profile to the output traffic at the interface set.

   ```
   [edit class-of-service interfaces interface-set interface-set-name]
   user@host# set output-traffic-control-profile profile-name
   ```

10. Apply the traffic control profile to the output traffic at each member interface of the ATM interface set.

   ```
   [edit class-of-service interfaces at-fpc/pic/port unit logical-unit-number]
   user@host# set output-traffic-control-profile profile-name
   ```

The following example configures ATM VP shaping on interface at-1/0/4 with VPI 40. The example defines an ATM interface set named atm-vp-ifset with two member ATM logical interfaces, at-1/0/4.50 and at-1/0/4.51, both of which use VPI 40. Traffic control profiles atm-vp-tcp1, atm-vp-tcp2, and atm-vp-tcp3 are each defined with the `atm-service`, `peak-rate`, `sustained-rate`, and `max-burst size` cell parameters. Finally, the `output-traffic-control-profile` statement applies traffic control profile atm-vp-tcp1 to the output traffic at interface at-1/0/4.50, atm-vp-tcp2 to the output traffic at interface at-1/0/4.51, and atm-vp-tcp3 to the output traffic at the atm-vp-ifset interface set.

```
[edit]
# Configure ATM Physical Interface
user@host# set interfaces at-1/0/4 hierarchical-scheduler
user@host# set interfaces at-1/0/4 atm-options vpi 40
#
# Configure ATM Logical Units
user@host# set interfaces at-1/0/4 unit 50 encapsulation pppoe-over-ether-over-atm-llc
user@host# set interfaces at-1/0/4 unit 50 vci 40.50
user@host# set interfaces at-1/0/4 unit 50 family pppoe duplicate-protection
user@host# set interfaces at-1/0/4 unit 51 encapsulation pppoe-over-ether-over-atm-llc
user@host# set interfaces at-1/0/4 unit 51 vci 40.51
user@host# set interfaces at-1/0/4 unit 51 family pppoe duplicate-protection
#
# Configure ATM Interface Set
user@host# set interfaces interface-set atm-vp-ifset interface at-1/0/4 unit 50
user@host# set interfaces interface-set atm-vp-ifset interface at-1/0/4 unit 51
#
# Configure Traffic Shaping and Scheduling Profiles
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set peak-rate 3k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set sustained-rate 200
user@host# set class-of-service traffic-control-profiles atm-vp-tcp1 set max-burst-size 1000
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set peak-rate 200
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set sustained-rate 100
user@host# set class-of-service traffic-control-profiles atm-vp-tcp2 set max-burst-size 150
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 atm-service nrtvbr
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set peak-rate 5k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set sustained-rate 1k
user@host# set class-of-service traffic-control-profiles atm-vp-tcp3 set max-burst-size 2000
#
# Apply Traffic Shaping and Scheduling Profiles
user@host# set class-of-service interfaces interface-set atm-vp-ifset output-traffic-control-
profile atm-vp-tcp3
```

```
user@host# set class-of-service interfaces at-1/0/4 unit 50 output-traffic-control-profile atm-
vp-tcp1
user@host# set class-of-service interfaces at-1/0/4 unit 51 output-traffic-control-profile atm-
vp-tcp2
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting in Junos OS Release 14.2, on MX Series routers with Modular Port Concentrator (MPC) interfaces and an ATM Modular Interface Card (MIC) with small form-factor pluggable transceiver (SFP) installed, you can configure class-of-service (CoS) hierarchical shaping and schedule for the traffic carried on an ATM virtual path (VP).

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421
Configuring CoS on Circuit Emulation ATM MICs
CoS on Circuit Emulation ATM MICs Overview

CHAPTER 35

Configuring Static Subscriber Interfaces over ATM

IN THIS CHAPTER

Example: Configuring a Static Subscriber Interface for IP Access over ATM

IN THIS SECTION

This example illustrates a routed IP-over-ATM (IPoA) configuration that creates a subscriber interface for a static IPv4 interface over a static ATM interface on an MX Series router. The router must have Module Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use an ATM MIC with small form-factor pluggable transceiver (SFP).

Requirements

This example uses the following software and hardware components:

- MX Series 5G Universal Routing Platform
- ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) and compatible MPC1 or MPC2

Before you begin:

1. Make sure the MX Series router you are using has an ATM MIC with SFP installed and operational.

   - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.

   - For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.

2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.

3. Define the static standard firewall filters (biz-customer-in-filter and biz-customer-out-filter) referenced in the configuration.

   - For information about creating standard firewall filters, see Guidelines for Configuring Firewall Filters.

   - For information about applying a firewall filter to an interface, see Guidelines for Applying Standard Firewall Filters.

Overview

By using the ATM MIC with SFP and a supported MPC, you can configure the MX Series router to support subscriber access for a statically created IPv4 or IPv6 interface over a static ATM underlying interface. An IPoA configuration enables you to provide access to subscribers on static IPv4 or IPv6 interfaces over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual circuits (PVCs).

> **NOTE**: IPoA configurations require static configuration of the IPv4 interface, IPv6 interface, CoS attributes, and firewall fiters. Dynamic configuration is not supported.

To configure IPoA subscriber access, configure the correct encapsulation type: `atm-snap` for IPoA encapsulation with logical link control (LLC), or `atm-vc-mux` for IPoA encapsulation with virtual circuit (VC) multiplexing. This example configures `atm-vc-mux` as the encapsulation type on the ATM logical interface.

To provision the ATM AAL5 PVCs for access over the ATM network, you must also configure the virtual path identifiers (VPIs) on the ATM physical interface, and one or more virtual circuit identifiers (VCIs) for each VPI.

In IPoA configurations, the subscriber interfaces correspond to the IPv4 or IPv6 addresses that are on the same network as the statically configured ATM underlying interface. In this IPoA example, the IPv4 address 10.0.0.2 represents the subscriber interface. You can configure the destination address with the set address 10.0.0.254/32 destination 10.0.0.2 statement at the [edit interfaces at-1/0/3 unit 0 family inet] hierarchy level.

This example includes the following basic steps to statically configure a single IPv4 subscriber interface over an ATM underlying interface:

1. Configure VPI 0 on ATM physical interface at-1/0/3.
2. Configure ATM VC multiplex encapsulation, VCI 0.39 (VCI 39 on VPI 0), and the following IPv4 (inet) protocol family characteristics on logical interface at-1/0/3.0 :
   - IP source address validation (rpf-check)
   - Standard input (biz-customer-in-filter) and output (biz-customer-out-filter) firewall filters
   - Interface address 10.0.0.254/32 with destination address 10.0.0.2
3. Configure static access route 10.200.10.0/24 with qualified-next-hop address at-1/0/0.0.

Configuration

IN THIS SECTION

To configure a static IPv4 subscriber interface over a static ATM underlying interface, perform these tasks:

CLI Quick Configuration

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

```
# ATM Physical Interface
set interfaces at-1/0/3 atm-options vpi 0
#
# Logical Unit 0
```

```
set interfaces at-1/0/3 unit 0 encapsulation atm-vc-mux
set interfaces at-1/0/3 unit 0 vci 0.39
set interfaces at-1/0/3 unit 0 family inet rpf-check
set interfaces at-1/0/3 unit 0 family inet filter input biz-customer-in-filter
set interfaces at-1/0/3 unit 0 family inet filter output biz-customer-out-filter
set interfaces at-1/0/3 unit 0 family inet address 10.0.0.254/32 destination 10.0.0.2
#
# Routing Properties
set routing-options access route 200.10.10.0/24 qualified-next-hop at-1/0/0.0
```

Configuring the ATM Physical Interface

Step-by-Step Procedure

To configure the ATM physical interface:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-1/0/3]
user@host# edit atm-options
```

2. Configure one or more VPIs on the physical interface.

```
[edit interfaces at-1/0/3 atm-options]
user@host# set vpi 0
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the ATM physical interface configuration by issuing the `show interfaces at-1/0/3` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/3
atm-options {
    vpi 0;
}
```

If you are done configuring the ATM physical interface, enter `commit` from configuration mode.

Configuring the Static IPv4 Subscriber Interface on Logical Unit 0

Step-by-Step Procedure

To configure the static IPv4 subscriber interface on logical unit 0:

1. Configure ATM VC multiplex encapsulation on the logical interface.

```
[edit interfaces at-1/0/3 unit 0]
user@host# set encapsulation atm-vc-mux
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/3 unit 0]
user@host# set vci 0.39
```

3. Configure the IPv4 (`inet`) protocol family, IPv4 address, and remote (destination) address of the connection.

```
[edit interfaces at-1/0/3 unit 0]
user@host# set family inet address 10.0.0.254/32 destination 10.0.0.2
```

4. Specify that you want to configure additional attributes for the IPv4 protocol family.

```
[edit interfaces at-1/0/3 unit 0]
user@host# edit family inet
```

5. Enable IP source address validation, which checks whether traffic is arriving at the router on an expected path.

```
[edit interfaces at-1/0/3 unit 0 family inet]
user@host# set rpf-check
```

6. Apply the previously defined standard firewall filters to the logical interface.

```
[edit interfaces at-1/0/3 unit 0 family inet]
user@host# set filter input biz-customer-in-filter
user@host# set filter output biz-customer-out-filter
```

Results

From the [edit] hierarchy level in configuration mode, confirm the results of the static subscriber interface configuration on logical unit 0 by issuing the show interfaces at-1/0/3.0 command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/3.0
encapsulation atm-vc-mux;
vci 0.39;
family inet {
    rpf-check;
    filter {
        input biz-customer-in-filter;
        output biz-customer-out-filter;
    }
    address 10.0.0.254/32 {
        destination 10.0.0.2;
    }
}
```

If you are done configuring the static subscriber interface on logical unit 0, enter commit from configuration mode.

Configuring Routing Properties

Step-by-Step Procedure

To configure static routing properties:

1. Specify that you want to configure protocol-independent routing properties.

```
[edit]
user@host# edit routing-options
```

2. Configure a static access route for routing downstream traffic from the router, and a qualified-next-hop address for routing upstream traffic to the router.

```
[edit routing-options]
user@host# set access route 200.10.10.0/24 qualified-next-hop at-1/0/0.0
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static routing properties configuration by issuing the `show routing-options` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show routing-options
access {
    route 200.10.10.0/24 {
        qualified-next-hop at-1/0/0.0;
    }
}
```

If you are done configuring the static routing properties, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the IPoA configuration is working properly, perform the following tasks:

Verifying the ATM Physical Interface Configuration

Purpose

Verify that the at-1/0/3 physical interface is properly configured for use with ATM PVCs.

Action

From operational mode, issue the `show interfaces at-1/0/3` command.

For brevity, this `show` command output includes only the configuration that is relevant to the at-1/0/3 physical interface. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show interfaces at-1/0/3
Physical interface: at-1/0/3, Enabled, Physical link is Down
  Interface index: 168, SNMP ifIndex: 595
  Link-level type: ATM-PVC, MTU: 2048, Clocking: Internal, SONET mode, Speed: OC3, Loopback:
None,
  Payload scrambler: Enabled
  Device flags   : Present Running Down
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 00:00:5e:00:53:18
  Last flapped   : 2012-08-28 07:14:48 PDT (08:28:47 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : LOL, LOS
  SONET defects  : LOL, LOS, LOP, BERR-SF, RDI-P
    VPI 0
      Flags: Active
      Total down time: 0 sec, Last down: Never
  Traffic statistics:
         Input  packets:                   0
         Output packets:                   0
  ...
```

Meaning

`ATM-PVC` in the Link-level Type field indicates that encapsulation for ATM permanent virtual circuits is being used on ATM physical interface at-1/0/3. The `Active` flag for VPI 0 indicates that the virtual path is up and operational.

Verifying the Static Subscriber Interface Configuration on Logical Unit 0

Purpose

Verify that the static subscriber interface on logical unit 0 is properly configured for IPv4 access over ATM.

Action

From operational mode, issue the `show interfaces at-1/0/3.0` command.

```
user@host> show interfaces at-1/0/3.0
  Logical interface at-1/0/3.0 (Index 341) (SNMP ifIndex 1984)
    Flags: Device-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: ATM-VCMUX
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: 2040
      Flags: Sendbcast-pkt-to-re, uRPF
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.0.0.2, Local: 10.0.0.254
    VCI 0.39
      Flags: Active
      Total down time: 0 sec, Last down: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`ATM-VCMUX` in the Encapsulation field indicates that the logical interface at-1/0/3.0 is properly configured for IPoA encapsulation with VC multiplexing. `Protocol inet` indicates that the IPv4 protocol family has been properly configured on the logical interface. The local address 10.0.0.254 is the IPv4 address of the logical interface. The destination address 10.0.0.2, which is in the same network as the local address, is the IPv4 address of the remote side of the connection and represents the static subscriber interface. The `Active` flag for VCI 0.39 indicates that virtual circuit identifier (VCI) 39 on VPI 0 is up and operational.

RELATED DOCUMENTATION

Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM

IN THIS SECTION

This example illustrates a bridged IP-over-Ethernet-over-ATM (IPoE-over-ATM) configuration that creates a subscriber interface for IPv4 access over a static ATM interface on an MX Series router. The router must have Module Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use an ATM MIC with small form-factor pluggable transceiver (SFP).

Requirements

This example uses the following software and hardware components:

- MX Series 5G Universal Routing Platform
- ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) and compatible MPC1 or MPC2

Before you begin:

1. Make sure the MX Series router you are using has an ATM MIC with SFP installed and operational.
    - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.

- For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.

2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.

3. Define the static standard firewall filters (biz-customer-in-filter and biz-customer-out-filter) referenced in the configuration.

   - For information about creating standard firewall filters, see Guidelines for Configuring Firewall Filters.

   - For information about applying a firewall filter to an interface, see Guidelines for Applying Standard Firewall Filters.

Overview

By using the ATM MIC with SFP and a supported MPC, you can configure the MX Series router to support subscriber access for a statically created IPv4 or IPv6 interface over a static ATM underlying interface. An IPoE-over-ATM configuration enables you to provide access to subscribers on static IPv4 or IPv6 interfaces over an underlying ATM interface on an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual circuits (PVCs).

> **NOTE**: IPoE-over-ATM configurations require static configuration of the IP interface, ATM interface, CoS attributes, and firewall fiters. Dynamic configuration is not supported.

To configure bridged IPoE-over-ATM subscriber access, you must configure Ethernet-over-ATM logical link control (LLC) encapsulation on the ATM underlying interface by including the `encapsulation ether-over-atm-llc` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level.

To provision the ATM AAL5 PVCs for access over the ATM network, you must also configure the virtual path identifiers (VPIs) on the ATM physical interface, and one or more virtual circuit identifiers (VCIs) for each VPI.

In IPoE-over-ATM configurations, the subscriber interfaces are associated with IPv4 or IPv6 addresses that are mapped to media access control (MAC) addresses. To statically configure Address Resolution Protocol (ARP) table entries that map IP address to MAC addresses, use the `arp` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet address` *`address`*`]` hierarchy level. In this example, the IPv4 address 10.0.50.2, configured with the `set arp 10.0.50.2 mac 00:00:5e:00:53:ff publish` statement at the `[edit interfaces at-1/0/2 unit 0 family inet address 10.0.50.254/24]` hierarchy level, represents the subscriber interface.

This example includes the following basic steps to statically configure a single IPv4 subscriber interface over an ATM underlying interface:

1. Configure VPI 0 on ATM physical interface at-1/0/2.
2. Configure Ethernet-over-ATM LLC encapsulation, VCI 0.39 (VCI 39 on VPI 0), and the following IPv4 (`inet`) protocol family characteristics on logical interface at-1/0/2.0 :
   - IPv4 subscriber interface address 10.0.50.254/24
   - Static Address Resolution Protocol (ARP) table entries that provide explicit mappings between IP addresses and MAC addresses
   - IP source address validation (`rpf-check`)
   - Standard input (biz-customer-in-filter) and output (biz-customer-out-filter) firewall filters
3. Configure static access route 200.10.10.0/24 with qualified-next-hop address at-1/0/0.0.

Configuration

IN THIS SECTION

To configure a static IPv4 subscriber interface over a static ATM underlying interface, perform these tasks:

CLI Quick Configuration

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

```
# ATM Physical Interface
set interfaces at-1/0/2 atm-options vpi 0
#
# Logical Unit 0
```

```
set interfaces at-1/0/2 unit 0 encapsulation ether-over-atm-llc
set interfaces at-1/0/2 unit 0 vci 0.39
set interfaces at-1/0/2 unit 0 family inet rpf-check
set interfaces at-1/0/2 unit 0 family inet filter input biz-customer-in-filter
set interfaces at-1/0/2 unit 0 family inet filter output biz-customer-out-filter
set interfaces at-1/0/2 unit 0 family inet address 10.0.50.254/24 arp 10.0.50.2 mac
00:00:5e:00:53:ff
set interfaces at-1/0/2 unit 0 family inet address 10.0.50.254/24 arp 10.0.50.2 publish
#
# Routing Properties
set routing-options access route 10.200.10.0/24 qualified-next-hop at-1/0/0.0
```

Configuring the ATM Physical Interface

Step-by-Step Procedure

To configure the ATM physical interface:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-1/0/2]
user@host# edit atm-options
```

2. Configure one or more VPIs on the physical interface.

```
[edit interfaces at-1/0/2 atm-options]
user@host# set vpi 0
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the ATM physical interface configuration by issuing the `show interfaces at-1/0/2` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/2
atm-options {
```

```
    vpi 0;
}
```

If you are done configuring the ATM physical interface, enter `commit` from configuration mode.

Configuring the Static IPv4 Subscriber Interface on Logical Unit 0

Step-by-Step Procedure

To configure the static IPv4 subscriber interface on logical unit 0:

1. Configure Ethernet-over-ATM LLC encapsulation on the logical interface.

   ```
   [edit interfaces at-1/0/2 unit 0]
   user@host# set encapsulation ether-over-atm-llc
   ```

2. Configure the VCI for the logical interface.

   ```
   [edit interfaces at-1/0/2 unit 0]
   user@host# set vci 0.39
   ```

3. Configure the IPv4 (`inet`) protocol family and address.

   ```
   [edit interfaces at-1/0/2 unit 0]
   user@host# set family inet address 10.0.50.254/24
   ```

4. Specify that you want to configure static ARP table entries to map between IP addresses and MAC addresses.

   ```
   [edit interfaces at-1/0/2 unit 0 family inet]
   user@host# edit family inet address 10.0.50.254/24
   ```

5. Configure IP address 10.0.50.2, which maps to the MAC address, and MAC address 00:00:5e:00:53:ff, which maps to the IP address. Include the `publish` option to specify that the router reply to ARP requests for the specified IP address.

```
[edit interfaces at-1/0/2 unit 0 family inet address 10.0.50.254/24]
user@host# set arp 10.0.50.2 mac 00:00:5e:00:53:ff publish
user@host# up
```

6. Enable IP source address validation, which checks whether traffic is arriving at the router on an expected path.

```
[edit interfaces at-1/0/2 unit 0 family inet]
user@host# set rpf-check
```

7. Apply the previously defined standard firewall filters to the logical interface.

```
[edit interfaces at-1/0/2 unit 0 family inet]
user@host# set filter input biz-customer-in-filter
user@host# set filter output biz-customer-out-filter
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static subscriber interface configuration on logical unit 0 by issuing the `show interfaces at-1/0/2.0` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/2.0
encapsulation ether-over-atm-llc;
vci 0.39;
family inet {
    rpf-check;
    filter {
        input biz-customer-in-filter;
        output biz-customer-out-filter;
    }
    address 10.0.50.254/24 {
        arp 10.0.50.2 mac 00:00:5e:00:53:ff publish;
```

```
    }
}
```

If you are done configuring the static subscriber interface on logical unit 0, enter `commit` from configuration mode.

Configuring Routing Properties

Step-by-Step Procedure

To configure static routing properties:

1. Specify that you want to configure protocol-independent routing properties.

   ```
   [edit]
   user@host# edit routing-options
   ```

2. Configure a static access route for routing downstream traffic from the router, and a qualified-next-hop address for routing upstream traffic to the router.

   ```
   [edit routing-options]
   user@host# set access route 10.200.10.0/24 qualified-next-hop at-1/0/0.0
   ```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static routing properties configuration by issuing the `show routing-options` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show routing-options
access {
    route 10.200.10.0/24 {
        qualified-next-hop at-1/0/0.0;
    }
}
```

If you are done configuring the static routing properties, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the IPoE-over-ATM configuration is working properly, perform the following tasks:

Verifying the ATM Physical Interface Configuration

Purpose

Verify that the at-1/0/2 physical interface is properly configured for use with ATM PVCs.

Action

From operational mode, issue the `show interfaces at-1/0/2` command.

For brevity, this `show` command output includes only the configuration that is relevant to the at-1/0/2 physical interface. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show interfaces at-1/0/2
Physical interface: at-1/0/2, Enabled, Physical link is Down
  Interface index: 175, SNMP ifIndex: 594
  Link-level type: ATM-PVC, MTU: 2048, Clocking: Internal, SDH mode, Speed: OC3, Loopback: None,
Payload scrambler: Enabled
  Device flags   : Present Running Down
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Current address: 00:00:5e:00:53:97
  Last flapped   : 2012-09-06 12:11:39 PDT (05:45:45 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SDH   alarms   : LOL, LOS
  SDH   defects  : LOL, LOS, LOP, BERR-SF, HP-FERF
    VPI 0
      Flags: Active
```

```
      Total down time: 0 sec, Last down: Never
  Traffic statistics:
         Input  packets:                    0
         Output packets:                    0
...
```

Meaning

ATM-PVC in the Link-level Type field indicates that encapsulation for ATM permanent virtual circuits is being used on ATM physical interface at-1/0/2. The Active flag for VPI 0 indicates that the virtual path is up and operational.

Verifying the Static Subscriber Interface Configuration on Logical Unit 0

Purpose

Verify that the static subscriber interface on logical unit 0 is properly configured for IPoE-over-ATM access.

Action

From operational mode, issue the `show interfaces at-1/0/2.0` command.

```
user@host> show interfaces at-1/0/2.0
  Logical interface at-1/0/2.0 (Index 336) (SNMP ifIndex 1983)
    Flags: Device-Down Point-To-Multipoint SNMP-Traps 0x4000 Encapsulation: Ether-over-ATM-LLC
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: 2016
      Flags: Sendbcast-pkt-to-re, uRPF
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.0.50/24, Local: 10.0.50.254, Broadcast: 10.0.50.255
    VCI 0.39
      Flags: Active, Multicast
      Total down time: 0 sec, Last down: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`Ether-over-ATM-LLC` in the Encapsulation field indicates that logical interface at-1/0/2.0 is properly configured for Ethernet-over-ATM encapsulation with LLC. `Protocol inet` indicates that the IPv4 protocol family has been properly configured on the logical interface. The destination address 10.0.50/24 identifies the network in which the subscriber interface (10.0.50.2) resides. The `Active` flag for VCI 0.39 indicates that virtual circuit identifier (VCI) 39 on VPI 0 is up and operational.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | 421
Configuring ATM for Subscriber Access | 430
Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM | 439
Example: Configuring a Static PPPoE Subscriber Interface over ATM | 453
Example: Configuring a Static Subscriber Interface for IP Access over ATM | 471
Example: Configuring a Static PPP Subscriber Interface over ATM | 489

Example: Configuring a Static PPP Subscriber Interface over ATM

IN THIS SECTION

This example illustrates a PPP-over-ATM (PPPoA) configuration that creates three static PPP logical subscriber interfaces over a static ATM underlying interface on an MX Series router. The router must have Module Port Concentrator/Modular Interface Card (MPC/MIC) interfaces that use an ATM MIC with small form-factor pluggable transceiver (SFP).

Requirements

This example uses the following software and hardware components:

- MX Series 5G Universal Routing Platform
- ATM MIC with SFP (Model Number MIC-3D-8OC3-2OC12-ATM) and compatible MPC1 or MPC2

Before you begin:

1. Make sure the MX Series router you are using has an ATM MIC with SFP installed and operational.
   - For information about compatible MPCs for the ATM MIC with SFP, see the *MX Series Interface Module Reference*.
   - For information about installing MPCs and MICs in an MX Series router, see the *Hardware Guide* for your MX Series router model.
2. Make sure you understand how to configure and use static ATM interfaces.

   See ATM Interfaces Overview.
3. Create the dynamic profile (pppoa-cos-profile) and access profile (pe-B-ppp-clients) referenced in the configuration.
   - For information about creating a basic dynamic profile, see *Configuring a Basic Dynamic Profile*.
   - For information about creating a dynamic profile for class of service (CoS) attributes, see *Configuring Traffic Scheduling and Shaping for Subscriber Access*.
   - For information about creating an access profile for PPP Challenge Handshake Authentication Protocol (CHAP) authentication, see *Configuring the PPP Challenge Handshake Authentication Protocol*.

Overview

By using the ATM MIC with SFP and a supported MPC, you can configure an MX Series router to support PPP subscriber access over an ATM network. PPPoA configurations on MX Series routers consist of one or more statically created PPP logical subscriber interfaces over a static ATM underlying interface.

Optionally, you can use dynamic profiles to dynamically or statically apply subscriber services services, such as CoS and firewall filters, to the static PPP logical interface. Configuring CoS and firewall filters in this manner enables you to efficiently and economically provide these services to PPP subscribers accessing the router over an ATM network using ATM Adaptation Layer 5 (AAL5) permanent virtual connections (PVCs). This example uses a previously configured dynamic profile named pppoa-cos-profile to apply traffic scheduling and shaping parameters to logical interface at-1/0/1.2.

To configure PPPoA subscriber access, configure the correct encapsulation type: `atm-ppp-llc` for PPPoA encapsulation with logical link control (LLC), or `atm-ppp-vc-mux` for PPPoA encapsulation with virtual circuit (VC) multiplexing. This example configures `atm-ppp-llc` as the encapsulation type on logical interface at-1/0/1.0, and `atm-ppp-vc-mux` as the encapsulation type on logical interfaces at-1/0/1.1 and at-1/0/1.2.

To provision the ATM AAL5 PVCs for access over the ATM network, you must also configure the virtual path identifiers (VPIs) on the ATM physical interface, and one or more virtual circuit identifiers (VCIs) for each VPI.

In PPPoA configurations, each statically configured logical interface (for example, at-1/0/1.0) corresponds to a PPP logical subscriber interface. This example configures three PPP logical subscriber interfaces over an ATM interface, as follows:

- The ATM physical interface (at-1/0/1) is statically configured with VPI 0 and VPI 2.
- Logical interface at-1/0/1.0 (logical unit 0) is configured with PPP-over AAL5 LLC encapsulation, VCI 0.120 (VCI 120 on VPI 0), PPP-specific options, and the IPv4 protocol family and address.
- Logical interface at-1/0/1.1 (logical unit 1) is configured with PPP-over-AAL5 VC multiplexing encapsulation, VCI 2.120 (VCI 120 on VPI 2), PPP-specific options, and the IPv4 protocol family and address.
- Logical interface at-1/0/1.2 (logical unit 2) is configured with PPP-over-AAL5 VC multiplexing encapsulation, VCI 2.121 (VCI 121 on VPI 2), PPP-specific options, and the IPv4 protocol family and address. The PPP-specific options include applying a dynamic profile named pppoa-cos-profile to the static PPP interface. The pppoa-cos-profile dynamic profile applies traffic scheduling and shaping parameters to the PPP logical subscriber interface.

Configuration

IN THIS SECTION

To configure static PPP logical subscriber interfaces over an ATM interface, perform these tasks:

CLI Quick Configuration

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

```
# ATM Physical Interface
set interfaces at-1/0/1 atm-options vpi 0
set interfaces at-1/0/1 atm-options vpi 2
#
# Logical Unit 0
set interfaces at-1/0/1 unit 0 encapsulation atm-ppp-llc
set interfaces at-1/0/1 unit 0 vci 0.120
set interfaces at-1/0/1 unit 0 ppp-options chap access-profile pe-B-ppp-clients
set interfaces at-1/0/1 unit 0 ppp-options chap local-name pe-A-at-1/0/1
set interfaces at-1/0/1 unit 0 keepalives interval 5
set interfaces at-1/0/1 unit 0 keepalives up-count 6
set interfaces at-1/0/1 unit 0 keepalives down-count 4
set interfaces at-1/0/1 unit 0 family inet address 192.0.2.133/30
#
# Logical Unit 1
set interfaces at-1/0/1 unit 1 encapsulation atm-ppp-vc-mux
set interfaces at-1/0/1 unit 1 vci 2.120
set interfaces at-1/0/1 unit 1 keepalives interval 6
set interfaces at-1/0/1 unit 1 keepalives up-count 6
set interfaces at-1/0/1 unit 1 keepalives down-count 4
set interfaces at-1/0/1 unit 1 family inet address 192.0.2.143/30
#
# Logical Unit 2
set interfaces at-1/0/1 unit 2 encapsulation atm-ppp-vc-mux
set interfaces at-1/0/1 unit 2 vci 2.121
set interfaces at-1/0/1 unit 2 ppp-options chap access-profile pe-A-ppp-clients
set interfaces at-1/0/1 unit 2 ppp-options chap local-name pe-A-at-1/0/1
set interfaces at-1/0/1 unit 2 ppp-options chap passive
set interfaces at-1/0/1 unit 2 ppp-options dynamic-profile pppoa-cos-profile
set interfaces at-1/0/1 unit 2 keepalives interval 5
set interfaces at-1/0/1 unit 2 keepalives up-count 6
set interfaces at-1/0/1 unit 2 keepalives down-count 4
set interfaces at-1/0/1 unit 2 family inet address 192.0.2.153/30
```

Configuring the ATM Physical Interface

Step-by-Step Procedure

To configure the ATM physical interface:

1. Specify that you want to configure ATM-specific options on the physical interface.

```
[edit interfaces at-1/0/1]
user@host# edit atm-options
```

2. Configure one or more VPIs on the physical interface.

```
[edit interfaces at-1/0/1 atm-options]
user@host# set vpi 0
user@host# set vpi 2
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the ATM physical interface configuration by issuing the `show interfaces at-1/0/1` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/1
atm-options {
    vpi 0;
    vpi 2;
}
```

If you are done configuring the ATM physical interface, enter `commit` from configuration mode.

Configuring the Static PPP Subscriber Interface on Logical Unit 0

Step-by-Step Procedure

To configure the static PPP subscriber interface on logical unit 0:

1. Configure PPP-over AAL5 LLC encapsulation on the logical interface.

```
[edit interfaces at-1/0/1 unit 0]
user@host# set encapsulation atm-ppc-llc
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/1 unit 0]
user@host# set vci 0.120
```

3. Specify that you want to configure options for PPP CHAP on the logical interface.

```
[edit interfaces at-1/0/1 unit 0]
user@host# edit ppp-options chap
```

4. Assign the previously configured pe-B-ppp-clients access profile to the PPP logical subscriber interface.

```
[edit interfaces at-1/0/1 unit 0 ppp-options chap]
user@host# set access-profile pe-B-ppp-clients
```

5. Configure the local name used by the interface in CHAP challenge and response packets.

```
[edit interfaces at-1/0/1 unit 0 ppp-options chap]
user@host# set local-name “pe-A-at-1/0/1”
user@host# up 2
```

6. Configure the transmission of keepalive messages on the logical interface.

```
[edit interfaces at-1/0/1 unit 0]
user@host# set keepalives interval 5
user@host# set keepalives up-count 6
user@host# set keepalives down-count 4
```

7. Configure the IPv4 (`inet`) protocol family and IP address.

```
[edit interfaces at-1/0/1 unit 0]
user@host# set family inet address 192.0.2.133/30
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static PPP subscriber interface configuration on logical unit 0 by issuing the `show interfaces at-1/0/1.0` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/1.0
encapsulation atm-ppp-llc;
vci 0.120;
ppp-options {
    chap {
        access-profile pe-B-ppp-clients;
        local-name pe-A-at-1/0/1;
    }
}
keepalives interval 5 up-count 6 down-count 4;
family inet {
    address 192.0.2.133/30;
}
```

If you are done configuring the PPP logical subscriber interface on logical unit 0, enter `commit` from configuration mode.

Configuring the Static PPP Subscriber Interface on Logical Unit 1

Step-by-Step Procedure

To configure the static PPP subscriber interface on logical unit 1:

1. Configure PPP-over-AAL5 VC multiplexing encapsulation on the logical interface.

```
[edit interfaces at-1/0/1 unit 1]
user@host# set encapsulation atm-ppc-vc-mux
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/1 unit 1]
user@host# set vci 2.120
```

3. Configure the transmission of keepalive messages on the logical interface.

```
[edit interfaces at-1/0/1 unit 1]
user@host# set keepalives interval 6
user@host# set keepalives up-count 6
user@host# set keepalives down-count 4
```

4. Configure the IPv4 (`inet`) protocol family and IP address.

```
[edit interfaces at-1/0/1 unit 1]
user@host# set family inet address 192.0.2.143/30
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static PPP subscriber interface configuration on logical unit 1 by issuing the `show interfaces at-1/0/1.1` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/1.1
encapsulation atm-ppp-vc-mux;
vci 2.120;
keepalives interval 6 up-count 6 down-count 4;
family inet {
    address 192.0.2.143/30;
}
```

If you are done configuring the PPP logical subscriber interface on logical unit 1, enter `commit` from configuration mode.

Configuring the Static PPP Subscriber Interface on Logical Unit 2

Step-by-Step Procedure

To configure the static PPP subscriber interface on logical unit 2:

1. Configure PPP-over-AAL5 VC multiplex encapsulation on the logical interface.

```
[edit interfaces at-1/0/1 unit 2]
user@host# set encapsulation atm-ppc-vc-mux
```

2. Configure the VCI for the logical interface.

```
[edit interfaces at-1/0/1 unit 2]
user@host# set vci 2.121
```

3. Specify that you want to configure options for PPP CHAP on the logical interface.

```
[edit interfaces at-1/0/1 unit 2]
user@host# edit ppp-options chap
```

4. Assign the previously configured pe-A-ppp-clients access profile to the PPP logical subscriber interface.

```
[edit interfaces at-1/0/1 unit 2 ppp-options chap]
user@host# set access-profile pe-A-ppp-clients
```

5. Configure the local name used by the interface in CHAP challenge and response packets.

```
[edit interfaces at-1/0/1 unit 2 ppp-options chap]
user@host# set local-name “pe-A-at-1/0/1”
```

6. Configure passive mode for CHAP authentication.

```
[edit interfaces at-1/0/1 unit 2 ppp-options chap]
user@host# set passive
user@host# up
```

7. Apply the previously configured pppoa-cos-profile dynamic profile to the PPP logical subscriber interface.

```
[edit interfaces at-1/0/1 unit 2 ppp-options]
user@host# set dynamic-profile pppoa-cos-profile
user@host# up
```

8. Configure the transmission of keepalive messages on the logical interface.

```
[edit interfaces at-1/0/1 unit 2]
user@host# set keepalives interval 5
user@host# set keepalives up-count 6
user@host# set keepalives down-count 4
```

9. Configure the IPv4 (`inet`) protocol family and IP address.

```
[edit interfaces at-1/0/1 unit 2]
user@host# set family inet address 192.0.2.153/30
```

Results

From the `[edit]` hierarchy level in configuration mode, confirm the results of the static PPP subscriber interface configuration on logical unit 2 by issuing the `show interfaces at-1/0/1.2` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit]
user@host# show interfaces at-1/0/1.2
encapsulation atm-ppp-vc-mux;
vci 2.121;
ppp-options {
    chap {
        access-profile pe-A-ppp-clients;
        local-name pe-A-at-1/0/1;
        passive;
    }
    dynamic-profile pppoa-cos-profile;
}
keepalives interval 5 up-count 6 down-count 4;
```

```
family inet {
    address 192.0.2.153/30;
}
```

If you are done configuring the PPP logical subscriber interface on logical unit 2, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the PPPoA configuration is working properly, perform the following tasks:

Verifying the ATM Physical Interface Configuration

Purpose

Verify that the at-1/0/1 physical interface is properly configured for use with ATM PVCs.

Action

From operational mode, issue the `show interfaces at-1/0/1` command.

For brevity, this `show` command output includes only the configuration that is relevant to the at-1/0/1 physical interface. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show interfaces at-1/0/1
Physical interface: at-1/0/1, Enabled, Physical link is Down
  Interface index: 166, SNMP ifIndex: 593
  Link-level type: ATM-PVC, MTU: 2048, Clocking: Internal, SONET mode, Speed: OC3, Loopback:
None, Payload scrambler: Enabled
  Device flags   : Present Running Down
  Link flags     : None
```

```
CoS queues     : 8 supported, 8 maximum usable queues
Schedulers     : 0
Current address: 00:00:5e:00:53:96
Last flapped   : 2012-06-29 15:35:29 PDT (2d 16:24 ago)
Input rate     : 0 bps (0 pps)
Output rate    : 0 bps (0 pps)
SONET alarms   : LOL, LOS
SONET defects  : LOL, LOS, LOP, BERR-SF, RDI-P
  VPI 0
    Flags: Active
    Total down time: 0 sec, Last down: Never
Traffic statistics:
       Input  packets:                   0
       Output packets:                   0
  VPI 2
    Flags: Active
    Total down time: 0 sec, Last down: Never
Traffic statistics:
       Input  packets:                   0
       Output packets:                   0

...
```

Meaning

`ATM-PVC` in the Link-level Type field indicates that encapsulation for ATM permanent virtual circuits is being used on ATM physical interface at-1/0/1. The `Active` flag for VPI 0 and VPI 2 indicates that these virtual paths are up and operational.

Verifying the Static PPPoA Configuration on Logical Unit 0

Purpose

Verify that the static PPP subscriber interface is properly configured on logical unit 0 (at-1/0/1.0).

Action

From operational mode, issue the `show interfaces at-1/0/1.0` command.

```
user@host> show interfaces at-1/0/1.0
Logical interface at-1/0/1.0 (Index 337) (SNMP ifIndex 1979)
```

```
    Flags: Device-Down Point-To-Point Inverse-ARP SNMP-Traps 0x4000 Encapsulation: ATM-PPP-LLC
    Input packets : 0
    Output packets: 0
  Keepalive settings: Interval 5 seconds, Up-count 6, Down-count 4
    LCP state: Down
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured, mpls: Not-
configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 2034
      Flags: Sendbcast-pkt-to-re, Protocol-Down
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 192.0.2.132/30, Local: 192.0.2.133, Broadcast: 192.0.2.135
    VCI 0.120
      Flags: Active, Inverse-ARP
      Total down time: 0 sec, Last down: Never
      ARP statistics
       Received: 0, Sent: 0, Denied: 0, Operation not supported: 0,
       Bad packet length: 0, Bad protocol: 0, Bad protocol length: 0,
       Bad hardware length: 0, Dropped: 0
       Last received: Never, Last sent: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`ATM-PPP-LLC` in the Encapsulation field indicates that logical interface at-1/0/1.0 is properly configured for PPP-over-AAL5 logical link control (LLC) encapsulation. `Protocol inet` indicates that the IPv4 protocol family has been properly configured on the logical interface. The `Active` flag for VCI 0.120 indicates that virtual circuit identifier (VCI) 120 on VPI 0 is up and operational.

Verifying the Static PPPoA Configuration on Logical Unit 1

Purpose

Verify that the static PPP subscriber interface is properly configured on logical unit 1 (at-1/0/1.1).

Action

From operational mode, issue the `show interfaces at-1/0/1.1` command.

```
user@host> show interfaces at-1/0/1.1

  Logical interface at-1/0/1.1 (Index 338) (SNMP ifIndex 1980)
    Flags: Device-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: ATM-PPP-VCMUX
    Input packets : 0
    Output packets: 0
  Keepalive settings: Interval 6 seconds, Up-count 6, Down-count 4
   LCP state: Down
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured, mpls: Not-
configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 2038
      Flags: Sendbcast-pkt-to-re, Protocol-Down
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 192.0.2.142/30, Local: 192.0.2.143, Broadcast: 192.0.2.145
    VCI 2.120
      Flags: Active, Inverse-ARP
      Total down time: 0 sec, Last down: Never
      ARP statistics
       Received: 0, Sent: 0, Denied: 0, Operation not supported: 0,
       Bad packet length: 0, Bad protocol: 0, Bad protocol length: 0,
       Bad hardware length: 0, Dropped: 0
       Last received: Never, Last sent: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`ATM-PPP-VCMUX` in the Encapsulation field indicates that the logical interface at-1/0/1.1 is properly configured for PPP-over-AAL5 VC multiplexing encapsulation. `Protocol inet` indicates that the IPv4 protocol family has been properly configured on the logical interface. The `Active` flag for VCI 2.120 indicates that virtual circuit identifier (VCI) 120 on VPI 2 is up and operational.

Verifying the Static PPPoA Configuration on Logical Unit 2

Purpose

Verify that the static PPP subscriber interface is properly configured on logical unit 2 (at-1/0/1.2).

Action

From operational mode, issue the `show interfaces at-1/0/1.2` command.

```
user@host> show interfaces at-1/0/1.2
Logical interface at-1/0/1.2 (Index 339) (SNMP ifIndex 1981)
    Flags: Device-Down Point-To-Point SNMP-Traps 0x4000 Encapsulation: ATM-PPP-VCMUX
    Input packets : 0
    Output packets: 0
  Keepalive settings: Interval 5 seconds, Up-count 6, Down-count 4
   LCP state: Down
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured, mpls: Not-
configured
  CHAP state: Closed
  PAP state: Closed
    Protocol inet, MTU: 2038
      Flags: Sendbcast-pkt-to-re, Protocol-Down
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 192.0.2.152/30, Local: 192.0.2.153, Broadcast: 192.0.2.155
    VCI 2.121
      Flags: Active
      Total down time: 0 sec, Last down: Never
        Input packets : 0
        Output packets: 0
```

Meaning

`ATM-PPP-VCMUX` in the Encapsulation field indicates that the logical interface at-1/0/1.2 is properly configured for PPP-over-AAL5 VC multiplexing encapsulation. `Protocol inet` indicates that the IPv4 protocol family has been properly configured on the logical interface. The `Active` flag for VCI 2.121 indicates that virtual circuit identifier 121 on VPI 2 is up and operational.

RELATED DOCUMENTATION

ATM for Subscriber Access Overview | **421**

Configuring ATM for Subscriber Access | **430**

Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM | **439**

Example: Configuring a Static PPPoE Subscriber Interface over ATM | **453**

Example: Configuring a Static Subscriber Interface for IP Access over ATM | **471**

Example: Configuring a Static Subscriber Interface for IP Access over Ethernet over ATM | **480**

CHAPTER 36

Verifying and Managing ATM Configurations

IN THIS CHAPTER

Verifying and Managing ATM Configurations for Subscriber Access

IN THIS SECTION

Purpose

View information about the static or dynamic subscriber interfaces configured over a static ATM underlying interface on an MX Series router with MPC/MIC interfaces and an ATM MIC with SFP.

Action

- To display information about the ATM physical interface to ensure that it is properly configured for use with ATM PVCs:

```
user@host> show interfaces at-fpc/pic/port
```

- To display information about the ATM logical interface to ensure that it is properly configured as a dynamic or static subscriber interface:

```
user@host> show interfaces at-fpc/pic/port.logical-unit-number
```

- To display information about all static PPPoE (`pp0`) subscriber interfaces for static PPPoE-over-ATM configurations:

```
user@host> show interfaces pp0
```

- To display information about a specified static PPPoE (`pp0`) subscriber interface for static PPPoE-over-ATM configurations:

```
user@host> show interfaces pp0.logical-unit-number
```

- To display detailed information about the PPPoE underlying interface for dynamic or static PPPoE-over-ATM configurations:

```
user@host> show pppoe underlying-interfaces at-fpc/pic/port.logical-unit-number detail
```

- To display extensive information, including packet statistics and lockout time settings, about the PPPoE underlying interface for dynamic or static PPPoE-over-ATM configurations:

```
user@host> show pppoe underlying-interfaces at-fpc/pic/port.logical-unit-number extensive
```

- To display extensive information about the active ATM subscriber with the specified ATM virtual path identifier (VPI) and ATM virtual circuit identifier (VCI):

```
user@host> show subscribers vpi vpi-identifier vci vci-identifier extensive
```

RELATED DOCUMENTATION

Configuring ATM for Subscriber Access | **430**

Example: Configuring a Dynamic PPPoE Subscriber Interface over ATM | **439**

Example: Configuring a Static PPPoE Subscriber Interface over ATM | **453**

6

PART

Troubleshooting

CHAPTER 37

Contacting Juniper Networks Technical Support

IN THIS CHAPTER

Collecting Subscriber Access Logs Before Contacting Juniper Networks Technical Support

IN THIS SECTION

Problem

Description

When you experience a subscriber access problem in your network, we recommend that you collect certain logs before you contact Juniper Networks 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 Networks 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 Networks 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 Networks Technical Support

CHAPTER 38

Knowledge Base

7

PART

Configuration Statements and Operational Commands

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

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

- Configuration Statements
- Operational Commands