IPv6 Multicast Over PPPoE

This chapter contains the following sections:

PPPoE in MANET

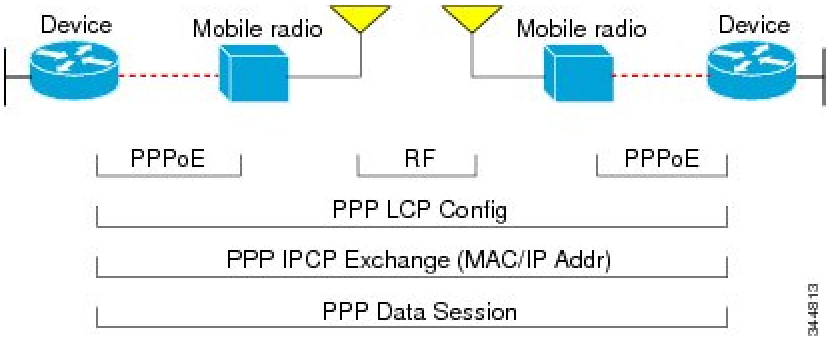
PPPoE combines Ethernet and PPP, to provide an authenticated method of assigning IP addresses to client systems.

The Cisco MANET solution employs PPPoE sessions to enable intra-nodal communications between a router and its partner radio. Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio. After the PPPoE sessions are active, a PPP session is established end-to-end (router-to-router).

This is duplicated each time a radio establishes a new radio link. The Virtual Multipoint Interface (VMI) on the device can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Underneath the VMI are virtual access interfaces that are associated with each of the PPP and PPPoE connections.

A PPPoE session is established between a device and a radio on behalf of every other device and radio neighbor located in the MANET. These Layer 2 sessions are the means by which radio network status gets reported to the Layer 3 processes in the device. The following figure shows the PPPoE session exchange between mobile devices and directional radios in a MANET network.

Figure 1: PPPoE Session Exchange Between Mobile Devices and Directional Radios



Link Control Protocol (LCP) forms part of the Point-to-Point Protocol (PPP), within the family of Internet protocols. The routers exchange IP Control Protocol (IPCP) messages to negotiate options specific to the protocol.

VMI Interface

The VMI on the device can aggregate all of the per-neighbor PPPoE sessions from the radio Ethernet connection. The VMI maps the sessions to appear to Layer 3 routing protocols and applications as a single point-to-multipoint, multiaccess, broadcast-capable network. However, the VMI preserves the integrity of the PPPoE sessions on the radio side so that each point-to-point connection can have its own quality of service (QoS) queue.

The VMI also relays the link-quality metric and neighbor up/down signaling from the radio to the routing protocols. The VMI signals are used by the Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv4 and IPv6 neighbors and the Open Shortest Path First version 3 (OSPFv3) for IPv6 neighbors.

The VMI can operate in two modes: bypass or aggregate.

Bypass Mode

This is the recommended mode for PPPOE in a MANET network.

In bypass mode, the virtual-access interfaces are directly exposed to applications running above L2. In bypass mode, you must still define a VMI because VMI continues to manage presentation of cross-layer signals, such as, neighbor up, neighbor down, and metrics. However, applications will still be aware of the actual underlying virtual-access interfaces and send packets to them directly.

If you are running multicast applications that require virtual-access interfaces to be exposed to applications above L2 directly, you can configure VMI to operate in bypass mode. Most multicast applications require that the virtual-access interfaces be exposed directly to routing protocols for the multicast Reverse Path Forwarding (RPF) to operate as expected .

Aggregate Mode

In this mode, all the virtual-access interfaces created by PPPoE sessions are aggregated logically under the configured VMI. VMI on the router can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Applications above Layer 2 (L2), such as Enhanced Interior Gateway Routing Protocol (EIGRP) and OSPFv3, should be defined only on VMI. Underneath VMI are virtual access interfaces that are associated with each of the PPP/PPPoE connections. Packets sent to VMI are forwarded to the correct virtual-access interface(s).

Aggregate mode VMIs operate in Non-Broadcast Multiple Access (NBMA) mode. Multicast traffic is forwarded only to the NBMA neighbors where a listener for that group is present. This is the preferred mode when operating in PIM sparse mode.

Virtual Access Interface

The Virtual-Access interfaces are logically “underneath” the VMI interface. Each Virtual-Access interface represents a “destination” which is either a routing next-hop, or a multicast group. At the bottom of the interface hierarchy is the actual physical interface connecting the router and radio. The Virtual-Access interface funnels the traffic to the physical interface for transmission to the radio device.

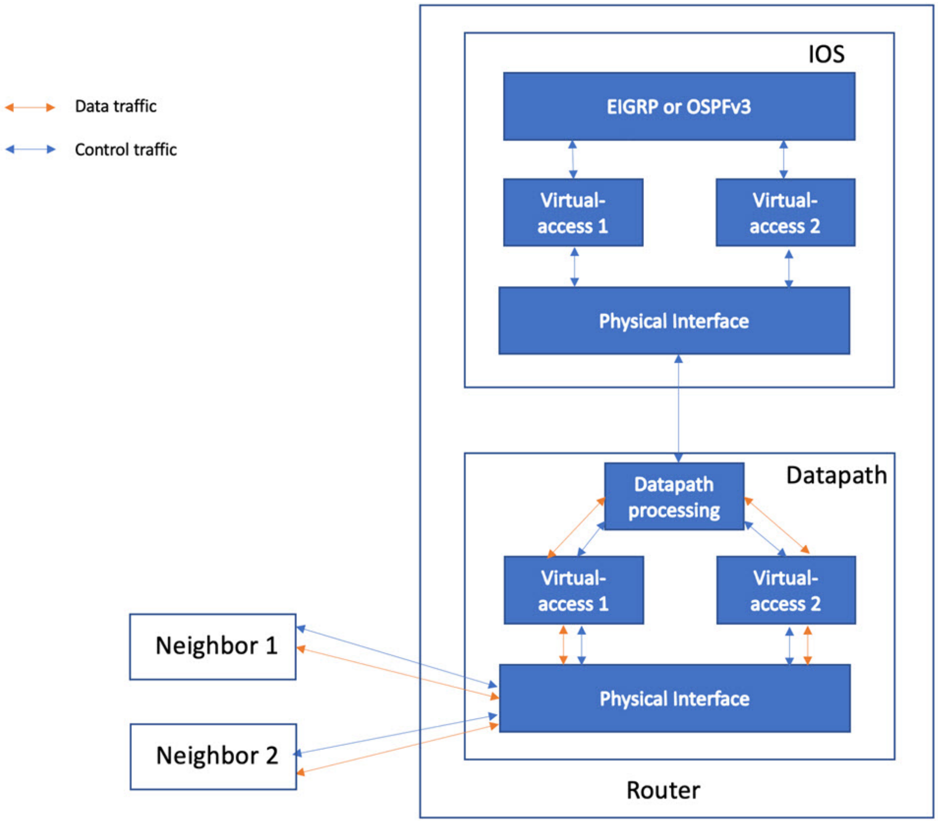
PPPOE Packet Flow

The Packet Flow diagram below illustrates the packet flow for both control and data packets over a PPPOE session when the VMI interface is in bypass mode.

All control traffic is sent to by the datapath to IOS where it is handed over to the appropriate protocol to be processed. The incoming interface for this traffic is the virtual-access interface associated with the neighbor which is the source of the traffic. In this case, Virtual-access1 corresponds to Neighbor 1 and Virtual-Access 2 corresponds to Neighbor 2.

All data traffic is processed by the datapath and does not typically get sent to IOS.

Figure 2: Packet Flow



Feature Requirements

For IPV6 multicast over PPPoE to function properly, the following must be configured:

- PPPoE (Virtual-template, VMI and physical interface)
- IPV6 unicast and multicast routing
- IPv6 PIM BSR
- IPv6 MLD

Note This feature requires the Network Advantage License.

Enabling IPv6 Routing

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**ipv6 unicast-routing** **Example:** Router(config)# `ipv6 unicast-routing`	Enable IPv6 Unicast routing.
Step 3	ipv6 multicast-routing **Example:** Router(config)#`ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces, and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.

Creating a Subscriber Profile

The Subscriber Profile Support feature is functionality for the Subscriber Service Switch architecture, a Cisco IOS subsystem that connects subscribers to network access services at Layer 2. This functionality affects how the Subscriber Service Switch Manager determines a service for each subscriber with a combination of a policy and a service lookup model.

Note Configuring a subscriber profile for PPPoE service selection is required for VMI to function properly.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**subscriber templating** **Example:** Router(config)#`subscriber templating`	Configure subscriber templating.
Step 3	subscriber authorization enable **Example:** Router(config)#`subscriber authorization enable`	Enable Subscriber Service Switch type authorization.

Configuring License Level

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**license boot level** *type* **Example:** Router(config)#`license boot level network-advantage`	Configure License information. **Note** Must be configured as Network Advantage for IPv6 multicast routing.

Configuring PPPoE Service Policy

A service policy specifies class-based weighted fair queueing (CBWFQ). The class policies that make up the policy map are then applied to packets that satisfy the class map match criteria for the class.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**policy-map type service** *name* **Example:** Router(config)#`policy-map type service pppoe_rar`	Configure policy map type.
Step 3	**pppoe service** *name* **Example:** Router(config-service-policymap)# `pppoe service manet_radio`	Configure service policy map. **Note** Enter the PPPoE service policy name as manet_radio.

Configuring QoS Provisioning

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**policy-map** *name* **Example:** Router(config)#`policy-map rar_policer`	Create a policy map name.
Step 3	**class** *criteria_name* **Example:** Router(config-pmap)# `class class-default`	Configure policy criteria as system default class.
Step 4	**police** *target_rate* **conform-action** *action* **exceed-action** *action* **Example:** Router(config-pmap-c)# `police 10000000 conform-action transmit exceed-action drop`	Configure QoS policy for the Virtual Template Interface.

Configuring PPPoE Service Selection

The PPPoE Service Selection feature uses service tags to enable a PPP over Ethernet (PPPoE) server to offer PPPoE clients a selection of services during call setup. The customer chooses one of the services offered, and the service is provided when the PPPoE session becomes active. This feature enables service providers to offer a variety of services and to charge customers according to the service chosen.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**bba-group pppoe** *name* **Example:** Router(config)# `bba-group pppoe rar_group_1`	Configure PPPoE global group.

	Command or Action	Purpose
Step 3	**virtual-template** *number* **Example:** Router(config-bba-group)# `virtual-template 1`	Attach the Virtual Template to the PPPoE bba group.
Step 4	**service profile** *name* **Example:** Router(config-bba-group)#`service profile pppoe_rar`	Attach the service policy name .

Configuring PPPoE on an Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**interface** *[type slot/port]* **Example:** Router(config)# `interface GigabitEthernet0/0/0`	Specifies an interface type and enters interface configuration mode.
Step 3	**ipv6 enable** **Example:** Router (config-if)# `ipv6 enable`	Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.
Step 4	**pppoe enable group** *group_name* **Example:** Router(config-if)#`pppoe enable group rar_group_1`	Enables PPPoE sessions on the interface or sub interface.

Configuring a Virtual Template Interface

You use the virtual template interface to dynamically clone configurations for each virtual access interface created for a virtual multipoint interface (VMI) neighbor. You can configure multiple virtual template interfaces for your VMI PPP over Ethernet (PPPoE) connections. The selection of which virtual template to use is predicated on the service name sent by the radio during PPPoE session establishment.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**interface Virtual-Template** *number* **Example:** Router(config)# `interface Virtual-Template 1`	Creates a Virtual Template Interface for configuration and dynamic application to Virtual Access Interfaces.
Step 3	**mtu** *size* **Example:** Router(config-if)# `mtu 1484`	Sets the MTU size.
Step 4	**ip unnumbered** *interface_type interface_number* **Example:** Router(config-if)#`ip unnumbered vmi 1`	Enable IP processing without an explicit address. Specifies the Virtual Template Interface to use the VMI interface IP address.
Step 5	**no ip redirects** **Example:** Router(config-if)#`no ip redirects`	Disables ICMP redirect messages.
Step 6	**ip tcp adjust-mss** *size* **Example:** Router(config-if)#`ip tcp adjust-mss 1444`	Adjust the mss of transit packets.
Step 7	**load-interval** *value* **Example:** Router(config-if)#`load-interval 30`	Load interval delay.
Step 8	**no peer default ip address** **Example:** Router(config-if)#`no peer default ip address`	Disables peer default ip address.
Step 9	**ipv6 enable** **Example:** Router(config-if)#`ipv6 enable`	Enable IPv6 support under Virtual Template Interface.
Step 10	**ipv6 mtu** *size* **Example:** Router(config-if)#`ipv6 mtu 1484`	Sets the IPv6 MTU size.
Step 11	**ospfv3 1 network** *type* **Example:**	Configure the Virtual Template Interface as MANET OSPF interface type.

	Command or Action	Purpose
	Router(config-if)#**ospfv3 1 network manet**	
Step 12	**ospfv3 1 hello-interval** *value* **Example:** Router(config-if)#**ospfv3 1 hello-interval 10**	Configure the hello interval value for OSPFv3.
Step 13	**ospfv3 1 ipv4 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv4 area 0**	Enable Virtual Template Interface to participate in OSPFv3 IPv4 routing.
Step 14	**ospfv3 1 ipv6 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv6 area 0**	Enable Virtual Template Interface to participate in OSPFv3 IPv6 routing.
Step 15	**service-policy input** *policy_map_name* **Example:** Router(config-if)#**service-policy input rar_policer**	Attach the created policy map name to the Virtual Template Interface.

Configuring VMI interface and Enabling Multicast Support

Used in router-to-radio communications, the Virtual Multipoint Interface (VMI) interface provides services that map outgoing packets to the appropriate Point-to-Point Protocol over Ethernet (PPPoE) sessions based on the next-hop forwarding address for that packet. The VMI interface also provides a broadcast service that emulates a set of point-to-point connections as a point-to-multipoint interface with broadcast ability. When a packet with a multicast address is forwarded through the VMI interface, VMI replicates the packet and unicasts it to each of its neighbors.

Note VMI will map outgoing packets to the appropriate PPPoE sessions. It will use the next-hop forwarding address from each outgoing packet to perform this mapping.

Note VMI is required to have IP addresses assigned for VMI to work even though it will be shown as down/down while in bypass mode. The IPv4 address configured will not be advertised or used. Instead, the IPv4 address on the virtual-template will be used.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Router# `configure terminal`	
Step 2	**interface vmi** *number* **Example:** Router(config)# `interface vmi 1`	Creates a VMI and enters interface configuration mode.
Step 3	**ipv6 enable** **Example:** Router(config-if)#`ipv6 enable`	Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.
Step 4	**ip address** *address mask* **Example:** Router(config-if)#`ip address 71.0.0.1 255.255.255.0`	Configures IPv4 address under VMI interface.
Step 5	**ipv6 address** *address/prefix* **Example:** Router(config-if)#`ipv6 address 71::71/64`	Configures IPv6 address under VMI interface.
Step 6	**physical-interface** *interface* **Example:** Router(config-if)#`physical-interface gigabitEthernet {0/0/0 or 0/0/1 or 0/0/0.1}`	Binds the physical interface (the interface connected to the radio client) to the VMI interface, for packet flow.
Step 7	**mode** *name* **Example:** Router(config-if)#`mode bypass`	Do not aggregate Virtual Access Interfaces under VMI. (Must be configured as bypass).

Configuring the Loopback Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**interface loopback** *number* **Example:** Router(config)# `interface loopback 1`	Assigns the loopback interface number.

	Command or Action	Purpose
Step 3	**ip address** *address mask* **Example:** Router(config-if)#`ip address 1.1.1.1 255.255.255.255`	Configuring IPv4 address for the loopback interface.
Step 4	**ipv6 address** *address/prefix* **Example:** Router(config-if)#`ipv6 address 11::11/128`	Configures IPv6 address for the loopback interface.
Step 5	**ipv6 enable** **Example:** Router(config-if)#`ipv6 enable`	Enable IPv6 support under the loopback interface.
Step 6	**ospfv3 1 ipv4 area** *id* **Example:** Router(config-if)#`ospfv3 1 ipv4 area 0`	Enable loopback interface to participate in OSPFv3 routing.
Step 7	**ospfv3 1 ipv6 area** *id* **Example:** Router(config-if)#`ospfv3 1 ipv6 area 0`	Enable loopback interface to participate in OSPFv3 routing.

Configuring the OSPFv3 IPv4 Address Family Process

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**router ospfv3** *process_id* **Example:** Router(config)# `router ospfv3 1`	Global configuration for OSPFv3.
Step 3	**router-id** *ipv4_address* **Example:** Router(config-router)#`router-id 101.101.101.101`	Router ID for OSPFv3.

	Command or Action	Purpose
Step 4	**address-family ipv4 unicast** **Example:** Router(config-router)#`address-family ipv4 unicast`	Adding address family for IPv4 unicast routing under global OSPFv3 configuration.
Step 5	**redistribute connected metric** *value* **metric-type** *type* **Example:** Router(config-router-af)#`redistribute connected metric 1 metric-type`	Redistribute metrics from external routing protocol.
Step 6	**log-adjacency-changes** **Example:** Router(config-if)#`log-adjacency-changes`	Logs all state changes.

Configuring the OSPFv3 IPv6 Address Family Process

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**router ospfv3** *process_id* **Example:** Router(config)# `router ospfv3 1`	Global configuration for OSPFv3.
Step 3	**router-id** *ipv4_address* **Example:** Router(config-router)#`router-id 101.101.101.101`	Router ID for OSPFv3.
Step 4	**address-family ipv6 unicast** **Example:** Router(config-router)#`address-family ipv6 unicast`	Adding address family for IPv6 unicast routing under global OSPFv3 configuration.
Step 5	**redistribute connected metric** *value* **metric-type** *type* **Example:**	Redistribute metrics from external routing protocol.

	Command or Action	Purpose
	Router(config-router-af)#`redistribute connected metric 1 metric-type`	
Step 6	**log-adjacency-changes** **Example:** Router(config-if)#`log-adjacency-changes`	Logs all state changes.

Configuring IPv6 PIM Bootstrap Router (BSR)

The bootstrap router (BSR) protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# `configure terminal`	Enters global configuration mode.
Step 2	**ipv6 pim bsr candidate bsr** *loopback_ipv6_address* **Example:** Router(config)# `ipv6 pim bsr candidate bsr 11::11`	Configures a device to be a candidate BSR. **Note** Must use the loopback ipv6 address configured earlier.
Step 3	**ipv6 pim bsr candidate rp** *loopback_ipv6_address* **Example:** Router(config-router)#`ipv6 pim bsr candidate rp 11::11`	Configures a device to be a candidate Rendezvous Point (RP). **Note** Must use the loopback ipv6 address configured earlier.

Configuring IPv6 Multicast Group

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Router# **`configure terminal`**	
Step 2	**interface** *interface_name* **Example:** Router# **`interface Vlan 10`**	Creates and enters interface configuration mode.
Step 3	**ipv6 mld join-group** *multicast_group_address* **Example:** Router(config-if)#**`ipv6 mld join-group FF06:6::1`**	Configure IPv6 mld join group under interface.

Verifying Virtual Template Interface

This section shows examples of command output to verify your setup.

```
Router# show vtemplate
Virtual access subinterface creation is globally enabled

         Active     Active    Subint  Interface
       Interface Subinterface Capable Type
       --------- ------------ ------- ---------
Vt1             0            1   Yes    Serial

Usage Summary
                              Interface   Subinterface
                              ---------   ------------
Current Serial  in use                2              1
Current Serial  free                  0              1
Current Ether   in use                0              0
Current Ether   free                  0              0
Current Tunnel  in use                0              0
Current Tunnel  free                  0              0
Current VPN  in use                   0              0
Current VPN  free                     0              0
Total                                 2              2

Cumulative created                    3             27
Cumulative freed                      0             26

Base virtual access interfaces: 2
Total create or clone requests: 3
Cancelled create or clone requests: 0
Cumulative create request waiting for sso resources: 0
Current request queue size: 0
Current free pending: 0
Current recycle pending: 0
Current ordered recycle pending: 0

Maximum request duration: 32 msec
Average request duration: 24 msec
Last request duration: 32 msec

Maximum processing duration: 32 msec
Average processing duration: 24 msec
Last processing duration: 32 msec
Router#
```

Verifying PPPoE Session Details

This section shows examples of command output to verify your setup.

```
Router-1#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1      000c.296f.c985  Gi0/0/0    1     Vi1.1    PTA    7c31.0e85.1e78
 UP
Router#

Router-2#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1      000c.29a1.ae42  Gi0/0/0    1     Vi1.1    PTA   d478.9b5d.0200
 UP
Router#

Router-1# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: 7c31.0e85.1e78, remote MAC address: 000c.296f.c985
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    67 packets sent, 67 received
    6488 bytes sent, 5908 received

PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 38     PADG Timer index: 89
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 89
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 36  rcvd: 0
 PADC xmit: 127926272  rcvd: 36
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 38, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 38, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 61
  PADC xmit: seq_num = 61, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-1#

Router-2# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: d478.9b5d.0200, remote MAC address: 000c.29a1.ae42
```

```
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    78 packets sent, 75 received
    7408 bytes sent, 6642 received

PPPoE Flow Control Stats
 Local Credits: 1950   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 42     PADG Timer index: 96
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 96
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 40  rcvd: 0
 PADC xmit: 127795200  rcvd: 40
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 42, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 42, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 66
  PADC xmit: seq_num = 63, fcn = 1950, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-2#
```

Verifying BSR Election

This section shows examples of command output to verify your setup.

```
Router-1#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
  This system is the Bootstrap Router (BSR)
     BSR Address: 61::61
     Uptime: 00:05:20, BSR Priority: 0, Hash mask length: 126
     RPF: FE80::7E31:EFF:FE85:1E78,Loopback0
     BS Timer: 00:00:41
  This system is candidate BSR
       Candidate BSR address: 61::61, priority: 0, hash mask length: 126
Router-1#
```

```
Router-2#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
     BSR Address: 61::61
     Uptime: 00:01:22, BSR Priority: 0, Hash mask length: 126
     RPF: FE80::7E31:EFF:FE85:1E78,Virtual-Access1.1
     BS Timer: 00:01:47
  This system is candidate BSR
       Candidate BSR address: 41::41, priority: 0, hash mask length: 126
Router-2#
```

IPv6 Multicast Routing Table

This section shows examples of command output to verify your setup.

Note “*” in the source address indicates that it is a mld join request entries.

```
Router-1#show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(A000::2, FF06:6::1), 00:01:45/00:01:44, flags: SFT
  Incoming interface: Vlan30
  RPF nbr: A000::2
  Immediate Outgoing interface list:
    Virtual-Access1.1, Forward, 00:01:45/00:02:48

Router-1#

Router-2# show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:6::1), 00:04:33/00:03:29, RP 41::41, flags: SCL
  Incoming interface: Tunnel2
  RPF nbr: 41::41
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:04:33/00:03:29

(A000::2, FF06:6::1), 00:01:41/00:02:43, flags: ST
  Incoming interface: Virtual-Access1.1
  RPF nbr: FE80::7E31:EFF:FE85:1E78
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:01:41/00:02:49
Router-2
```

IPv6 Multicast over PPPoE

This new feature applies to the ESR6300 Router.

PPPoE is a session/connection-oriented protocol, which extends the point-to-point radio frequency (RF) link from an external radio to an IOS router. Router communication with the radio is represented by virtual access interface (connectivity to a radio neighbor).

VMI operates in the Bypass mode where each Virtual Access Interface (VAI) represents a radio neighbor. The VMI layer re-directs unicast routing protocol traffic to the appropriate P2P link (Virtual-Access interface) and replicates any Multicast traffic that needs to flow.

For IPV6 multicast over PPPoE to function properly, the following must be configured:

- PPPoE (Virtual-template, VMI and physical interface)
- IPV6 unicast and multicast routing
- IPv6 PIM BSR
- IPv6 MLD

Note This feature requires the Network Advantage License.

For additional information, see the IPv6 Multicast over PPPoE chapter.

Verifying VMI Neighbors

This section shows examples of command output to verify your setup.

The Multicast for Virtual Multipoint Interfaces feature enables multicast support for RFC 5578-compliant Radio-Aware Routing (RAR). Multicast is defined as a network group membership spanning the entire network. The virtual multipoint interface (VMI) operates in aggregate mode, which means that all virtual access interfaces created by PPP over Ethernet (PPPoE) sessions are aggregated logically under the configured VMI. Packets sent to the VMI are forwarded to the correct virtual access interface. When a VMI operates in aggregate mode, the interfaces operate in nonbroadcast multiple access (NBMA) mode. Multicast traffic is forwarded only the he NBMA neighbors where a listener for that group is present.

```
Router-1#show vmi neighbor detail
              1 vmi1 Neighbors


vmi1    IPV6 Address=FE80::D678:9BFF:FE5D:200
       IPV6 Global Addr=::
       IPV4 Address=81.0.0.1, Uptime=00:03:16
       Output pkts=0, Input pkts=0
       METRIC DATA: Total rcvd=2, Avg arrival rate (ms)=10220
          CURRENT: MDR=2000000000000 bps, CDR=1000000000 bps
                   Lat=1 ms, Res=100, RLQ=90, load=1
          MDR      Max=2000000000000 bps, Min=2000000000000 bps, Avg=2000000000000 bps
          CDR      Max=1000000000 bps, Min=1000000000 bps, Avg=1000000000 bps
          Latency  Max=1, Min=1, Avg=1 (ms)
          Resource Max=100%, Min=100%, Avg=100%
          RLQ      Max=90, Min=90, Avg=90
          Load     Max=1%, Min=0%, Avg=0%
       Transport PPPoE, Session ID=2
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
 Local Credits: 1954   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 198     PADG Timer index: 283
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 283
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 196  rcvd: 0
 PADC xmit: 128057344  rcvd: 196
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 198, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 198, fcn = 1954, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 105
  PADC xmit: seq_num = 109, fcn = 1951, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 1
Router-1#

Router#

Router-2#show vmi neighbor detail
              1 vmi1 Neighbors


vmi1    IPV6 Address=FE80::7E31:EFF:FE85:1E78
       IPV6 Global Addr=::
       IPV4 Address=71.0.0.1, Uptime=00:01:50
       Output pkts=0, Input pkts=0
       No Session Metrics have been received for this neighbor.
       Transport PPPoE, Session ID=1
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65533   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 112     PADG Timer index: 168
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 168
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 110  rcvd: 0
 PADC xmit: 127991808  rcvd: 110
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 112, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 112, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 70
  PADC xmit: seq_num = 71, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0

Router-2#
```

Verifying OSPF Neighbor

This section shows examples of command output to verify your setup.

```
Router-1#sh ospfv3 neighbor | i FULL
102.102.102.102   0   FULL/  -        00:00:33    24    Virtual-Access1.1
102.102.102.102   0   FULL/  -        00:00:34    24             Virtual-Access1.1
Router-1
```

```
Router-2# sh ospfv3 neighbor | i FULL
sh ospfv3 neighbor | i FULL
101.101.101.101   0   FULL/  -        00:00:33    23             Virtual-Access1.1
101.101.101.101   0   FULL/  -        00:00:33    23             Virtual-Access1.1
Router-2#
```

Sample Running Configuration

The following output shows a sample configuration with areas of interest highlighted.

```
Router-1#show run
Building configuration...

Current configuration : 7911 bytes
!
! Last configuration change at 13:04:28 UTC Wed Jun 8 2022
!
version 17.9
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 250M
platform punt-keepalive disable-kernel-core
!
hostname Router-1
!
boot-start-marker
boot system flash bootflash:c6300-universalk9.17.09.01.SSA.bin
boot-end-marker
!
!
no logging console
no aaa new-model
!
```

```
!
login on-success log
ipv6 unicast-routing
ipv6 multicast-routing
!
!
subscriber templating
subscriber authorization enable
!
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint TP-self-signed-4073554590
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-4073554590
 revocation-check none
 rsakeypair TP-self-signed-4073554590
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-4073554590
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 34303733 35353435 3930301E 170D3232 30363038 31313137
  31345A17 0D333230 36303731 31313731 345A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 30373335
  35343539 30308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100C54E 860F73AC F0A9EBC7 2C6D1204 49099324 85989550 32CA9B91
  3B3A2492 AEA1D550 0CD787DE 09F6B64F 3F01C578 3EFB3995 E448904A 957EAFD9
  B82EF201 3A28BE26 6B1615DA 35B35BEA D4B7B20F 2D2A3EC2 C1F52281 349E88E5
  F2BCBA37 CA72D461 97D1E817 7493ED38 7C7C1035 F7231D4E F59FADF9 EE0EE5EE
  1FD73691 E93EBE8C 262DC8B1 0FA25BF0 C2F65BF9 C57A406A 9F9CF3D3 3E6888C3
  D6B533AB 0DA71037 6C94A385 CECA4DD9 A037C344 5B761E6D F3B8D47B 4093BED8
  E497D649 63436773 7BE5A718 331C7F08 31071542 03AE588C 08605CF7 CC7C7D6F
  967759FC E2A943FF 8AD70094 825AAD4B DC66FEC8 5B7F2CDA 24F148E1 51AF106E
  FE212C21 651B0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14F49EA0 C5E78BB1 F0EA55AD 4C580FB7 BF1AE35C
  50301D06 03551D0E 04160414 F49EA0C5 E78BB1F0 EA55AD4C 580FB7BF 1AE35C50
  300D0609 2A864886 F70D0101 05050003 82010100 2083EBC1 E760806B 12F0ACEF
  8FDE2E11 E5B88B62 B85B5835 AB1D2471 EA6FCC2D B28F5EE2 1969C233 DBB8C435
  D7BAD49C 7781E485 97D5B5D0 DD05A0EE 5352535F 1657BE78 64E6BBA8 B627618E
  49041DFB 4FE4D16C FA6857EF E6EEDFBC 2E25AF9E 852EDF71 3B65E55F 62AB1B1E
  8B842F51 DAD55DB5 8A5BF87B 91F540D2 02E8576E 5D4550EA E7FCB6D7 6AD0E92A
  EEAF7544 01C4095A BB02DB3A D45D73EB 971974FF B5DED058 F2F3A0E1 23BD3441
  899CDFC2 A3B36E7D E72D4BDB 480B8347 C26D6AB9 E7E5A140 B20B7B1D 7AC24A2C
  A69124B3 49BF18AF C99EF2A5 C4F484CE 9E2A70C1 D1EA4250 6E0D858E BBCFC6C1
  4FF6E0BC EB190067 E86EC80D 5D149D6F 462CA857
   quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
   quit
!
!
no license feature hseck9
license udi pid ESR-6300-CON-K9 sn FOC234304H3
license boot level network-advantage
memory free low-watermark processor 45135
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
 mode none
!
!
vlan internal allocation policy ascending
!
policy-map type service pppoe_rar
 pppoe service manet_radio
!
!
bba-group pppoe rar_group_1
 virtual-template 1
 service profile pppoe_rar
!
!
interface Loopback0
 ip address 61.0.0.1 255.255.255.255
 ipv6 address 61::61/128
 ipv6 enable
 ospfv3 1 ipv6 area 0
 ospfv3 1 ipv4 area 0
!
interface GigabitEthernet0/0/0
 media-type rj45
 negotiation auto
 ipv6 enable
 pppoe enable group rar_group_1
!
interface GigabitEthernet0/0/1
```

```
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 30
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Virtual-Template1
 mtu 1484
 ip unnumbered vmi1
 no ip redirects
 ip tcp adjust-mss 1444
 load-interval 30
 no peer default ip address
 ipv6 enable
 ipv6 mtu 1484
 ospfv3 1 network manet
 ospfv3 1 hello-interval 10
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
!
interface Vlan1
 no ip address
!
interface Vlan30
 ip address 192.168.10.1 255.255.255.0
 ipv6 address A000::1/64
 ipv6 enable
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
interface vmi1
 ip address 71.0.0.1 255.255.255.0
 ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
 ipv6 address 71::71/64
 ipv6 enable
 physical-interface GigabitEthernet0/0/0
 mode bypass
!
router ospfv3 1
 router-id 101.101.101.101
 !
 address-family ipv4 unicast
  redistribute connected metric 1 metric-type 1
  log-adjacency-changes
 exit-address-family
 !
 address-family ipv6 unicast
  redistribute connected metric-type 1
  log-adjacency-changes
 exit-address-family
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
ipv6 pim bsr candidate bsr 61::61
ipv6 pim bsr candidate rp 61::61
!
!
control-plane
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 stopbits 1
line 0/0/0
line 0/2/0
line vty 0 4
 login
 transport input ssh
line vty 5 14
 login
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
!

end
```

Debug Commands

This section shows debug commands for PPPoE, VMI, Virtual Template, and Subscriber Service.

PPPoE

Command or Action	Purpose
Router# **debug pppoe data**	PPPoE data packets
Router# **debug pppoe errors**	PPPoE protocol errors
Router# **debug pppoe events**	PPPoE protocol events
Router# **debug pppoe packets**	PPPoE control packets

VMI

Command or Action	Purpose
Router# **debug vmi bma**	Display VMI bma debug
Router# **debug vmi error**	Display internal VMI anomalies detected
Router# **debug vmi multicast**	Display VMI multicast packets
Router# **debug vmi neighbor**	Display VMI neighbor transaction debugging
Router# **debug vmi packet**	Display VMI packet trace
Router# **debug vmi pppoe**	Display VMI PPPoE packet/activity debug
Router# **debug vmi registries**	Display VMI registry calls

Virtual Template

Command or Action	Purpose
Router# **debug vtemplate clonin**g	Virtual Template cloning information
Router# **debug vtemplate error**	Virtual Template errors
Router# **debug vtemplate event**	Virtual Template events
Router# **debug vtemplate subinterface**	Virtual Template subinterface command

Subscriber Service

Command or Action	Purpose
Router# **debug sss errors**	Subscriber Service Switch Manager errors
Router# **debug ssss event**	Subscriber Service Switch Manager events