



CHAPTER 8

Configuring PPPoE

After configuring the interfaces and verifying connectivity as described in [Chapter 3, “Configuring the Interfaces,”](#) the next step is configuring the protocols.

Prerequisite Reading

Read the following chapter before selecting a RAR protocol:

- [Chapter 5, “Introduction to Radio Aware Routing and MANET”](#)



Note

You can use only one RAR protocol per interface.

This chapter contains the following sections:

- [PPPoE in a MANET, page 8-1](#)
- [VMI in a MANET, page 8-2](#)
- [PPPoE and VMI, page 8-6](#)
- [Configuring PPPoE for use with VMI, page 8-7](#)
- [Showing VMI Neighbors, page 8-18](#)

PPPoE in a MANET

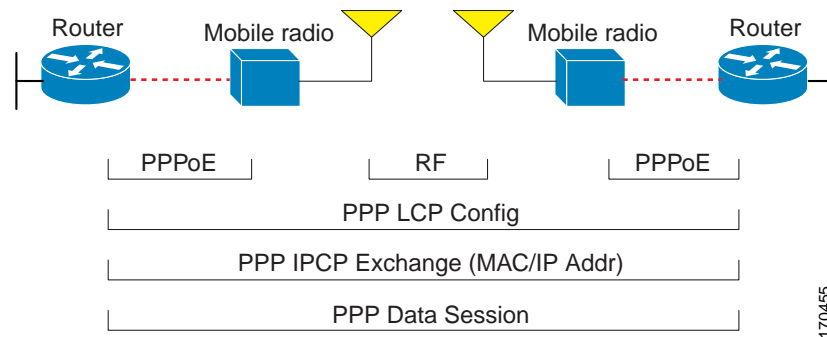
The Cisco MANET solution employs PPPoE sessions to enable intra-nodal communications between a router and its partner radio. Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio. After the PPPoE sessions are active, a PPP session is established end-to-end (router-to-router.) This is duplicated each time a radio establishes a new radio link. VMI on the router can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Underneath VMI are virtual access interfaces that are associated with each of the PPP/PPPoE connections.

If you are running multicast applications that require the 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 the routing protocols to ensure that multicast Reverse Path Forwarding (RPF) can operate as expected. When you use the bypass mode, you must define a VMI to handle presentation of cross-layer signals such as neighbor up, neighbor down, and metrics. Applications are aware of the actual underlying virtual-access interfaces and send

packets to the underlying virtual-access interfaces directly. Additional information is required on the virtual template configuration. Operating VMI in bypass mode can cause databases in the applications to be larger than would normally be expected because knowledge of more interfaces is required for normal operation.

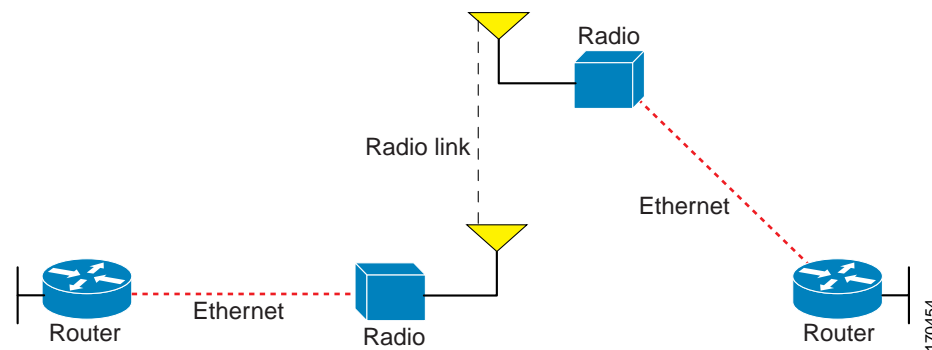
A PPPoE session is established between a router and a radio on behalf of every other router/radio neighbor located in the MANET. These L2 sessions are the means by which radio network status gets reported to the Layer 3 (L3) processes in the router. [Figure 8-1](#) illustrates the PPPoE session exchange between mobile routers and directional radios in a MANET.

Figure 8-1 PPPoE Session Exchange Between Mobile Routers and Directional Radios



This capability requires that an RFC-5578 compliant radio be connected to a router using Ethernet. The router always considers the Ethernet link to be up. If the radio side of the link goes down, the router waits until a routing update time-out occurs to declare that the route is down and then updates the routing table. [Figure 8-2](#) illustrates a simple router-to-radio link topology. The routing protocols optimized for VMI PPPoE are EIGRP (IPv4, IPv6) and OSPFv3 (IPv4, IPv6).

Figure 8-2 Router-to-Radio Link



VMI in a MANET

VMI provides services that map outgoing packets to the appropriate PPPoE sessions based on the next-hop forwarding address for that packet. VMI 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 VMI in aggregate mode, VMI replicates the packet and sends it using the virtual-access interface(s) to each of its neighbors.

Directional radios are frequently used in applications that require greater bandwidth, increased power-to-transmission range, or reduced probability of detection. These radios operate in a point-to-point mode, and generally have no broadcast capability. On the other hand, the routing processes in Cisco's MANET solution operate most efficiently when viewing the network link as point-to-multipoint, with broadcast capability. For the router, modeling the MANET as a collection of point-to-point nodes has a dramatic impact on the size of its internal database.

VMI within the router can aggregate all of the per-neighbor PPPoE sessions from the Radio Ethernet connection. VMI maps the sessions to appear to L3 routing protocols and applications as a single point-to-multipoint, multi-access, broadcast-capable network. However, 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.

VMI also relays the link quality metric and neighbor up/down signaling from the radio to the routing protocols. Currently, VMI signals are used by Enhanced Interior Gateway Routing Protocol (EIGRP) (for IPv4 and IPv6 neighbors) and OSPFv3 (for IPv6 neighbors).

Link-Quality Metrics

The quality of a radio link has a direct impact on the throughput. The PPPoE protocol has been extended to provide a process by which a router can request report link quality metric information. Cisco's OSPFv3 and EIGRP implementations are enhanced so that the route cost to a neighbor is dynamically updated based on metrics reported by the radio, thus allowing the best route to be chosen within a given set of radio links.

The routing protocols receive raw radio link data, and compute a composite quality metric for each link. In computing these metrics, the router may consider the following factors:

- Maximum Data Rate—the theoretical maximum data rate of the radio link, in scaled bits per second
- Current Data Rate—the current data rate achieved on the link, in scaled bits per second
- Latency—the transmission delay packets encounter, in milliseconds
- Resources—a percentage (0-100) that can represent the remaining amount of a resource (such as battery power)
- Relative Link Quality—a numeric value (0-100) representing relative quality, with 100 being the highest quality

On the router, metrics can be weighted during the configuration process to emphasize or de-emphasize particular characteristics. For example, if throughput is a particular concern, you can weight the *throughput* metric so that it is factored more heavily into the composite route cost. Similarly, a metric of no concern can be omitted from the composite calculation.

Link metrics can change rapidly, often by very small degrees, which could result in a flood of meaningless routing updates. In a worst case scenario, the network churns almost continuously as it struggles to react to minor variations in link quality. To alleviate this concern, Cisco provides a tunable dampening mechanism that allows the user to configure threshold values. Any metric change that falls below the threshold is ignored. The quality of a connection to a neighbor varies, based on various characteristics of the interface when OSPFv3 or EIGRP is used as the routing protocol. The routing protocol receives dynamic raw radio link characteristics and computes a composite metric that is used to reduce the effect of frequent routing changes.

A tunable hysteresis mechanism allows you to adjust the threshold to the routing changes that occur when the router receives a signal that a new peer has been discovered, or that an existing peer is unreachable. The tunable metric is weighted and adjusted dynamically to account for the following characteristics:

- Current and Maximum Bandwidth
- Latency
- Resources
- Relative Link Quality (RLQ)

Individual weights can be deconfigured and all weights can be cleared so that the cost returns to the default value for the interface type. Based on the routing changes that occur, cost can be determined by the application of these metrics.

Neighbor Signaling

MANETs are highly dynamic environments. Nodes may move into, or out of, radio range at a fast pace. Each time a node joins or leaves the network, topology must be logically reconstructed by the routers. Routing protocols normally use timer-driven “hello” messages or neighbor time-outs to track topology changes, but MANETs reliance on these mechanisms can result in unacceptably slow convergence.

Neighbor up/down signaling capability provides faster network convergence by using link-status signals generated by the radio. The radio notifies the router each time a link to another neighbor is established or terminated by the creation and termination of PPPoE sessions. In the router, the routing protocols (OSPFv3 or EIGRP) respond immediately to these signals by expediting the formation of a new adjacency (for a new neighbor) or tearing down an existing adjacency (if a neighbor is lost). For example, if a vehicle drives behind a building and loses its connection, the router immediately senses the loss and establishes a new route to the vehicle through neighbors that are not blocked. This high speed network convergence is essential for minimizing dropped voice calls and disruptions to video sessions.

When VMI with PPPoE is used and a partner node has left or a new one has joined, the radio informs the router immediately of the topology change. Upon receiving the signal, the router immediately declares the change and updates the routing tables.

The signaling capability provides the following benefits:

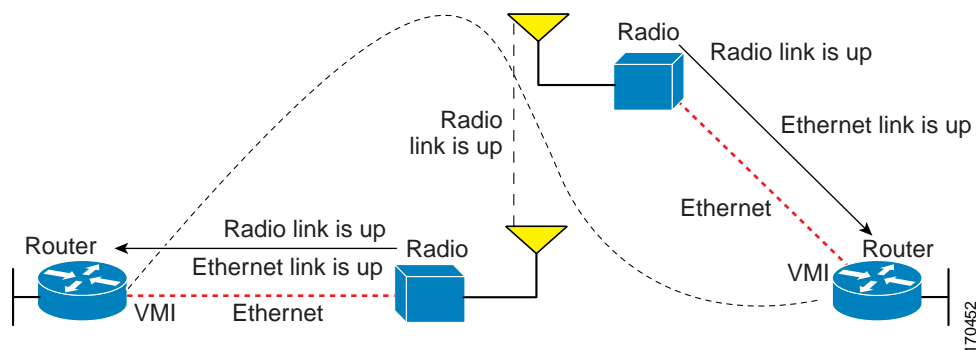
- Reduces routing delays and prevents applications from timing out
- Enables network-based applications and information to be delivered reliably and quickly over directional radio links
- Provides faster convergence and optimal route selection so that delay-sensitive traffic such as voice and video are not disrupted
- Reduces impact on radio equipment by minimizing the need for internal queuing/buffering
- Provides consistent Quality of Service (QoS) for networks with multiple radios

The messaging allows for flexible rerouting when necessary because of the following conditions:

- Noise on the Radio links
- Fading of the Radio links
- Congestion of the Radio links
- Radio link power fade
- Utilization of the Radio

Figure 8-3 illustrates the signaling sequence that occurs when radio links go up and down.

Figure 8-3 Up and Down Signaling Sequence



PPPoE Credit-based Flow Control

Each radio initiates a PPPoE session with its local router as soon as the radio establishes a link to another radio. Once the PPPoE sessions are active for each node, a PPP session is then established end-to-end (router-to-router). This process is duplicated each time a radio establishes a new link.

The carrying capacity of each radio link may vary due to location changes or environmental conditions, and many radio transmission systems have limited buffering capabilities. To minimize the need for packet queuing in the radio, Cisco has implemented extensions to the PPPoE protocol that enable the router to control traffic buffering in congestion situations. Implementing flow-control on these router-to-radio sessions also allows the use of fair queuing.

The flow control solution utilizes a credit-granting mechanism documented in RFC 5578. When the PPPoE session is established, the radio can request a flow-controlled session. If the router acknowledges the request, all subsequent traffic must be flow-controlled. If a flow control session has been requested and cannot be supported by the router, the session is terminated. Typically, both the radio and the router initially grant credits during session discovery. Once a device exhausts its credits, it must stop sending until additional credits have been granted. Credits can be added incrementally over the course of a session.

High performance radios that require high-speed links use metrics scaling. The radio can express the maximum and current data rates with different scaler values. Credit scaling allows a radio to change the default credit grant (or scaling factor) of 64 bytes to its default value. You can view the maximum and current data rates and the scalar value set by the radio from the output of the **show vmi neighbor detail** command.

Point-to-Point Protocol over Ethernet

Cross-layer feedback for router-radio integration radio aware routing takes advantage of the functions defined in RFC 5578. RFC 5578 is an Internet Engineering Task Force (IETF) standard that defines Point-to-Point Protocol over Ethernet (PPPoE) extensions for Ethernet-based communications between a router and a device such as a mobile radio that operates in a variable-bandwidth environment and has limited buffering capabilities. These extensions provide a PPPoE session based mechanism for sharing radio network status such as link-quality metrics and establishing flow control between a router and an RFC 5578-capable radio.

An RFC 5578 radio initiates an L2 PPPoE session with its adjacent router on behalf of every router and radio neighbor discovered in the network. These L2 sessions are the means by which radio network status for each neighbor link is reported to the router. The radio establishes correspondence between each PPPoE session and each link to a neighbor.

PPPoE and VMI

To use the PPPoE and Virtual Multipoint Interface (VMI) features described in this document, a radio device that implements the PPPoE functionality described in the RFC 2516 and RFC 5578 is required. OSPF enhancements are not tied to the PPPoE/VMI implementations, and as such do not require such radio devices.

VMI provides services that map outgoing packets to the appropriate PPPoE sessions based on the next-hop forwarding address for that packet. VMI 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 VMI in aggregate mode, VMI replicates the packet and sends it using the virtual-access interface(s) to each of its neighbors.



Note

VMI operates in aggregate mode by default. This release supports VMI in aggregate mode and also in bypass mode.

Directional radios are frequently used in applications that require greater bandwidth, increased power-to-transmission range, or reduced probability of detection. These radios operate in a point-to-point mode, and generally have no broadcast capability.

Conversely, the routing processes in the Cisco MANET solution operate most efficiently when viewing the network link as point-to-multipoint with broadcast capability. For the router, modeling the MANET as a collection of point-to-point nodes has a dramatic impact on the size of its internal database.

VMI within the router can aggregate all of the per-neighbor PPPoE sessions from the radio Ethernet connection. VMI maps the sessions to appear to L3 routing protocols and applications as a single point-to-multipoint, multi-access, broadcast-capable network. However, 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.

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

Continuing with PPPoE Configuration

This chapter provides the following major sections to describe how to configure Point-to-Point Protocol over Ethernet (PPPoE) on a specific interface.

- [Configuring PPPoE for use with VMI, page 8-7](#)
- [Showing VMI Neighbors, page 8-18](#)



Note

See [Appendix A, “Command Reference”](#) for detailed command reference.

Configuring PPPoE for use with VMI

This section provides the tasks required to configure PPPoE for use with Virtual Multipoint Interface (VMI):

- [Creating a Subscriber Profile, page 8-7](#)
- [Configuring PPPoE Service Selection, page 8-8](#)
- [Configuring PPPoE on an Ethernet Interface, page 8-9](#)
- [Configuring a Virtual Template Interface, page 8-10](#)
- [Mapping Outgoing Packets, page 8-12](#)
- [Configuring Multicast Support, page 8-14](#)

Creating a Subscriber Profile

Perform this task to configure a subscriber profile for PPPoE service selection.



Note

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **exit**
4. **subscriber authorization enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<code>exit</code> Example: Router(config-sss-profile)# <code>exit</code>	Returns to global configuration mode.
Step 4	<code>subscriber authorization enable</code> Example: Router# <code>subscriber authorization enable</code>	Enable Subscriber Service Switch type authorization. This command is required when VPDN is not used.

Configuring PPPoE Service Selection

Perform this task to associate the subscriber profile with a PPPoE profile. In this configuration, the Broadband Access (BBA) group name must match the subscriber profile name defined in the subscriber profile.



Note In this example, *manet_radio* serves as the subscriber profile name.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `bba-group pppoe {group-name | global}`
4. `virtual-template template-number`
5. `service profile subscriber-profile-name [refresh minutes]`
6. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>bba-group pppoe {group-name global}</code> Example: Router(config)# <code>bba-group pppoe pppoe_group_1</code>	Defines a PPPoE profile and enters BBA group configuration mode.

	Command or Action	Purpose
Step 4	<code>virtual-template template-number</code> Example: Router(config-bba-group)# <code>virtual-template 1</code>	Specifies the virtual template required for cloning virtual-access interfaces. All PPPoE ports using this PPPoE profile will use this virtual template.
Step 5	<code>service profile subscriber-profile-name</code> [<code>refresh minutes</code>] Example: Router(config-bba-group)# <code>service profile subscriber_1</code>	Assigns a subscriber profile to a PPPoE profile. <ul style="list-style-type: none"> The PPPoE server will advertise the service names that are listed in the subscriber profile to each PPPoE client connection that uses the configured PPPoE profile.
Step 6	<code>end</code> Example: Router(config-bba-group)# <code>end</code>	(Optional) Returns to privileged EXEC mode.

Configuring PPPoE on an Ethernet Interface

Perform this task to assign a PPPoE profile to an Ethernet interface.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface [type slot/port]`
4. `pppoe enable [group group-name]`
5. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<pre>interface [type slot/port]</pre> <p>Example: Router(config)# interface fastethernet 1/0</p>	<p>Specifies an interface type and enters interface configuration mode. Valid interfaces include the following interface types:</p> <ul style="list-style-type: none"> • Fast Ethernet interface • Ethernet • Fast Ethernet • Gigabit Ethernet • VLAN or VLAN subinterface
Step 4	<pre>pppoe enable [group group-name]</pre> <p>Example: Router(config-if)# pppoe enable group pppoe_group_1</p>	<p>Enables PPPoE sessions on the interface or subinterface.</p>
Step 5	<pre>end</pre> <p>Example: Router(config-if)# end</p>	<p>(Optional) Exits the configuration mode and returns to privileged EXEC mode.</p>

Configuring a Virtual Template Interface

Perform this task to configure a virtual-template interface. The virtual-template interface is required to clone configurations. For each VMI neighbor, a new virtual-access interface will be created dynamically.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no virtual-template subinterface**
4. **policy-map** *policy-map-name*
5. **class class-default**
6. **fair-queue**
7. **exit**
8. **interface virtual-template 1**
9. **ip unnumbered vmi1**
10. **service-policy output FQ**
11. **keepalive 60 20**
12. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>no virtual template subinterface</code> Example: Router# <code>no virtual template subinterface</code>	Disables the virtual template on the subinterface.
Step 4	<code>policy-map policy-map-name</code> Example: Router(config-pmap)# <code>policy-map FQ</code>	Enters policy map configuration mode and creates, or modifies, a policy map that can be attached to one or more interfaces to specify a service policy.
Step 5	<code>class class-default</code> Example: Router(config-pmap)# <code>class class-default</code>	Specifies one of the following: <ul style="list-style-type: none">Class Name for the policy you are about to create or changeDefault Class (also known as a <i>Class-default Class</i>) for general policy configuration
Step 6	<code>fair-queue</code> Example: Router(config-pmap)# <code>fair-queue</code>	Enables Weighted Fair Queueing (WFQ) in the policy-map.
Step 7	<code>exit</code> Example: Router(config-pmap)# <code>exit</code> Router(config)#	Exits the current mode and returns to configuration mode.
Step 8	<code>interface virtual-template number</code> Example: Router(config)# <code>interface virtual-template 1</code>	Creates a virtual-template interface for configuration and dynamic application to virtual-access interfaces.
Step 9	<code>ip unnumbered interface-type interface-number</code> Example: Router(config-if)# <code>ip unnumbered vm11</code>	Enables IP processing of IPv4 on the interface without assigning an explicit IP address.
Step 10	<code>service-policy output policy-map-name</code> Example: Router(config-if)# <code>service-policy output FQ</code>	Attaches a policy map to an input interface, Virtual Circuit (VC), or output interface. This policy map will serve as the service policy for that interface or VC.

	Command or Action	Purpose
Step 11	keepalive [[keepalive-period] [keepalive-retries]] Example: Router(config-if)# keepalive 60 20	Enables a keepalive period of 60 seconds with 20 retries.
Step 12	end Example: Router(config-if)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Example Configuration

```

no virtual-template subinterface
!
policy-map FQ
  class class-default
    fair-queue
!
interface Virtual-Template1
  ip unnumbered vmi1
  keepalive 60 20
  service-policy output FQ
!end

```

Mapping Outgoing Packets

Perform this task so that VMI can map outgoing packets to the appropriate PPPoE sessions. VMI will use the next-hop forwarding address from each outgoing packet perform this mapping.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vmi** *interface-number*
4. **ip address** *ip_addr subnet_mask*
5. **physical-interface** *interface-type/slot*
6. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>interface vmi number</code> Example: Router(config)# <code>interface vmi1</code>	Creates a VMI interface.
Step 4	<code>ip address ip_addr subnet_mask</code> Example: Router(config-if)# <code>ip address 10.2.2.1 255.255.255.0</code>	Specifies the IP address and subnet mask for the VMI interface.
Step 5	<code>physical-interface interface-type/slot</code> Example: Router(config-if)# <code>physical-interface fa0/0</code>	Creates the physical subinterface to be associated with VMI on the router.
Step 6	<code>end</code> Example: Router(config-if)# <code>end</code>	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Examples

The following examples show the IP address coordination needed between virtual-template configuration and VMI configuration.

VMI in Aggregate Mode for IPv6

The following example shows the configuration of VMI in aggregate mode for IPv6.

```
interface Virtual-Template1
  ipv6 enable
  service-policy output FQ
!
interface vmi1
  ipv6 enable
  physical-interface FastEthernet0/0
!
```

VMI in Aggregate Mode for IPv4

The following example shows the configuration of VMI in aggregate mode for IPv4.

```
interface Virtual-Template1
  ip unnumbered vmi1
  service-policy output FQ
!
interface vmi1
  ip address 10.2.2.1 255.255.255.0
  physical-interface FastEthernet0/0
!
```

VMI in Aggregate Mode for IPv4 and IPv6

The following example shows the configuration of VMI in aggregate mode for IPv4 and IPv6.

```
interface Virtual-Template1
  ip unnumbered vmi1
  ipv6 enable
  service-policy output FQ
!
interface vmi1
  ip address 10.2.2.1 255.255.255.0
  ipv6 enable
  physical-interface FastEthernet0/0
!
```

Configuring Multicast Support

This section identifies the recommended modes and tasks for working with multicast:

- [Using Aggregate Mode, page 8-14](#)
- [Using Bypass Mode, page 8-16](#)
- [Enabling Multicast Support on a VMI, page 8-16](#)

Using Aggregate Mode

VMI operates in aggregate mode by default. All of the virtual-access interfaces created by PPPoE sessions are aggregated logically under the configured VMI. Applications above Layer 2 (L2), such as Enhanced Interior Gateway Routing Protocol (EIGRP) and OSPFv3, should be defined only on VMI. 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.



Note

NBMA multicasting only supports IPv4 and sparse mode.


Perform this task to configure interface vmi1 to operate in NBMA mode and PIM sparse mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vmi** *interface-number*

4. **ip address** *ip_addr subnet_mask*
5. **ip pim nbma-mode**
6. **ip pim sparse-mode**
7. **load-interval** *number*
8. **physical-interface** *interface-type/slot*
9. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface vmi number Example: Router(config)# interface vmi1	Creates a VMI interface.
Step 4	ip address ip_addr subnet_mask Example: Router(config-if)# ip address 10.2.2.2 255.255.255.0	Specifies the IP address and subnet mask for the VMI interface. This example sets the IP address to 10.2.2.2 and the subnet mask to 255.255.255.0
Step 5	ip pim nbma-mode	Enables NBMA mode.
Step 6	ip pim sparse-mode	Enables sparse mode.
		 Note You must set this to sparse mode.
Step 7	load interval seconds Example: Router(config-if)# load-interval 30	Specifies the load interval in seconds. This example sets the load interval to 30 seconds.
Step 8	physical-interface interface-type/slot Example: Router(config-if)# physical-interface fa0/0	Creates the physical subinterface to be associated with VMI on the router.
Step 9	end Example: Router(config-if)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Using Bypass Mode

Using bypass mode is recommended for multicast applications.

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.

Using bypass mode can cause databases in the applications to be larger because knowledge of more interfaces are required for normal operation.

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 in order for the multicast Reverse Path Forwarding (RPF) to operate as expected. When you use the bypass mode, you must define a VMI to handle cross-layer signals such as neighbor up, neighbor down, and metrics. Applications will be aware of the actual underlying virtual-access interfaces, and will send packets to them directly. Operating VMI in bypass mode can cause databases in the applications to be larger than normally expected because knowledge of more interfaces is required for normal operation.

Enabling Multicast Support on a VMI

Perform this task to enable bypass mode on a VMI and override the default aggregation that occurs on VMI. This configuration assumes that you have already configured a virtual template and appropriate PPPoE sessions for VMI.

After you enter the enable bypass mode, Cisco recommends that you copy the running configuration to Non-Volatile Random Access Memory (NVRAM) because the default mode of operation for VMI is to logically aggregate the virtual-access interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *vmi number*
4. **mode bypass**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<code>interface vmi number</code> Example: Router(config-if)# <code>interface vmi1</code>	Enters interface configuration mode and relates a VMI interface.
Step 4	<code>mode bypass</code> Example: Router(config-if)# <code>mode bypass</code>	Overrides the default aggregation on the VMI interface and sets the mode to bypass to support multicast traffic on the interface.
Step 5	<code>exit</code> Example: Router(config-if)# <code>exit</code> Router(config)#	Exits the current mode.

Examples



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.

VMI in Bypass Mode for IPv6

The following example shows the configuration of VMI in bypass mode for IPv6.

```
interface Virtual-Template1
  ipv6 enable
  service-policy output FQ
!
interface vmi1
  ipv6 enable
  mode bypass
  physical-interface FastEthernet0/0
!
```

VMI in Bypass Mode for IPv4

The following example shows the configuration of VMI in bypass mode for IPv4.



Note

The IPv4 address configured on VMI will not be advertised or used. Instead, the IPv4 address on the virtual-template will be used.

```
interface Virtual-Template1
  ip address 10.1.1.1 255.255.255.0
  service-policy output FQ
!
interface vmi1
  ip address 2.2.2.1 255.255.255.0
  mode bypass
  physical-interface FastEthernet0/0
!
```

VMI in Bypass Mode for IPv4 and IPv6

The following example shows the configuration of VMI in bypass mode for IPV4 and IPv6.

```
interface Virtual-Template1
 ip address 10.1.1.1 255.255.255.0
 ipv6 enable
 service-policy output FQ
!
interface vmi1
 ip address 2.2.2.1 255.255.255.0
 ipv6 enable
 mode bypass
 physical-interface FastEthernet0/0
!
```

Showing VMI Neighbors

To display information about neighbor connections to VMI, use the **show vmi neighbors** command in User EXEC mode.

The following example shows how to display neighbors created dynamically on a VMI:

```
Router# show vmi neighbors vmi1

1 vmi1 Neighbors

      Interface      IPV6      IPV4      Uptime      Transmit      Receive
      Address        Address   Address   Uptime      Packets       Packets
vmi1      ::          10.3.3.2  00:02:11   0000000008   0000000073
Router#
```

Example

The following example shows the details about known VMI neighbors.

```
Router# show vmi neighbors detail

      1 vmi1 Neighbors

vmi1  IPV6 Address=FE80::A8BB:CCFF:FE00:C00
      IPV4 Address=12.12.12.2, Uptime=00:12:19
      Output pkts=0, Input pkts=0
      METRIC DATA: Total rcvd=3, Avg arrival rate (ms)=234952
      CURRENT: MDR=2048000, CDR=1024000, Lat=70, Res=100, RLQ=95, load=1
      MDR      Max=10240000, Min=2048000, Avg=4795050
      CDR      Max=10240000, Min=1024000, Avg=4104192
      Latency  Max=1000, Min=70, Avg=380
      Resource Max=100, Min=100, Avg=100
      RLQ      Max=100, Min=95, Avg=96
      Load     Max=1, Min=1, Avg=1
      Transport PPPoE, Session ID=1
      INTERFACE STATS:
      VMI Interface=vmi1,
      Input qcount=0, drops=0, Output qcount=0, drops=0
      V-Access intf=Virtual-Access2,
      Input qcount=0, drops=0, Output qcount=0, drops=0
      Physical intf=Ethernet0/0,
      Input qcount=0, drops=0, Output qcount=0, drops=0

      PPPoE Flow Control Stats
```

```
Local Credits: 65296   Peer Credits: 65196
Credit Grant Threshold: 28000   Max Credits per grant: 65534
PADG Seq Num: 696   PADG Timer index: 0
PADG last rcvd Seq Num: 697
PADG last nonzero Seq Num: 0
PADG last nonzero rcvd amount: 0
PADG Timers:   [0]-1000   [1]-2000   [2]-3000   [3]-4000
PADG xmit: 698   rcvd: 698
PADC xmit: 698   rcvd: 698
PADQ xmit: 0   rcvd: 2
Router#
```

