



Radio Aware Routing PPPoE

Radio-Aware Routing (RAR) is a mechanism that uses radios to interact with the routing protocols to signal the appearance, disappearance, and link conditions of routing neighbors.

- [Feature Information for RAR PPPoE, on page 1](#)
- [Radio Aware Routing PPPoE Overview, on page 2](#)
- [Restrictions, on page 7](#)
- [Enabling IPv6 Routing, on page 8](#)
- [Creating a Subscriber Profile, on page 8](#)
- [Configuring PPPoE Service Policy, on page 9](#)
- [Configuring QoS Provisioning, on page 9](#)
- [Configuring PPPoE Service Selection, on page 10](#)
- [Configuring PPPoE on an Ethernet Interface, on page 10](#)
- [Configuring a Virtual Template Interface, on page 11](#)
- [Configuring the Loopback Interface, on page 13](#)
- [Configuring the OSPFv3 IPv4 Address Family Process , on page 13](#)
- [Configuring the OSPFv3 IPv6 Address Family Process , on page 14](#)
- [Verifying Virtual Template Interface, on page 15](#)
- [Verifying PPPoE Session Details, on page 16](#)
- [Verifying VMI Neighbors, on page 17](#)
- [Verifying OSPF Neighbor, on page 19](#)

Feature Information for RAR PPPoE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for RAR PPPoE

Feature Name	Releases	Feature Information
IPv6 Multicast over RAR PPPoE	Cisco IOS XE Release 17.11.1a	This feature was introduced for the following platforms: <ul style="list-style-type: none"> • Cisco Catalyst 8000V Edge • Cisco Catalyst 8200 Series Edge Platforms • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 Series Edge Platforms
RAR PPPoE	Cisco IOS XE Release 17.8.1a	This feature was introduced for the following platforms: <ul style="list-style-type: none"> • Cisco Catalyst 8000V Edge
	Cisco IOS XE Release 16.6	This feature was introduced for the following platforms: <ul style="list-style-type: none"> • Cisco 4000 Series ISRs

Radio Aware Routing PPPoE Overview

This section provides a high-level description on how RAR, MANETs, and PPPoE work together.

About MANETs

Mobile Ad Hoc Networks (MANETs) for device-to-radio communications address the challenges faced when merging IP routing and mobile radio communications in ad hoc networking applications.

Through the device-to-radio link, the radio can inform the device immediately when a node joins or leaves, and this enables the device to recognize topology changes more quickly than if it had to rely on timers. Without this link-status notification from the radio, the device would likely time out while waiting for traffic. The link-status notification from the radio enables the device to respond faster to network topology changes. Metric information regarding the quality of a link is passed between the device and radio, enabling the device to more intelligently decide on which link to use.

With the link-status signaling provided by the device-to-radio link, applications such as voice and video work better because outages caused by topology changes are reduced or eliminated. Sessions are more stable and remain active longer.

PPPoE Extensions

PPPoE extensions are used when the router communicates with the radio. PPPoE is a well-defined communication mechanism between the client and the server. In the RAR implementation, radio takes the role of the PPPoE client and router takes the role of the PPPoE server. This allows a loose coupling of radio and router, while providing a well-defined and predictable communication mechanism. As PPPoE is a session or a connection oriented protocol, it extends the point-to-point radio frequency (RF) link from an external radio to an IOS-XE router.

Cross-layer feedback for device-to-radio integration of Radio-Aware Routing (RAR) takes advantage of the functions defined in RFC 5578. The RFC 5578 is an Internet Engineering Task Force (IETF) standard that defines PPP over Ethernet (PPPoE) extensions for Ethernet-based communications between a device and a mobile radio, that operates in a variable-bandwidth environment and has limited buffering capabilities. These extensions provide a PPPoE session-based mechanism for real time sharing of radio network status & link-quality metrics, and support credit-based flow control between router and RAR-compliant radio.

An RAR-compliant radio initiates a Layer 2 PPPoE session with its adjacent device on behalf of every device and radio neighbor discovered in the network. These Layer 2 sessions are the means by which radio network status for each neighbor link is reported to the device. The radio establishes the correspondence between each PPPoE session and each link to a neighbor.

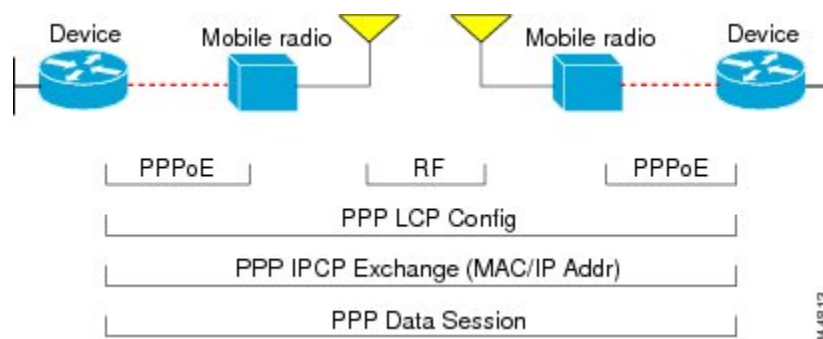
In the Cisco IOS implementation of PPPoE, each individual session is represented by virtual access interface (connectivity to a radio neighbor) on which, QoS can be applied with these PPPoE extensions.

PPPoE Interfaces for Mobile Radio Communications

The Mobile Ad Hoc Network (MANET) implementation uses PPP over Ethernet (PPPoE) sessions to enable intranodal communications between a device 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 (device-to-device). 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.

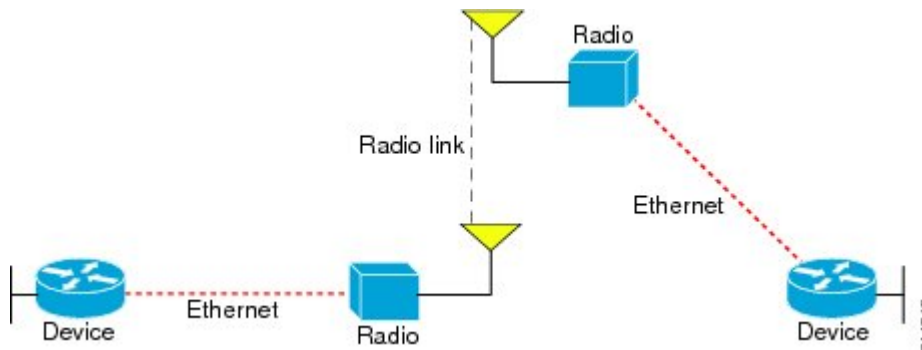
A PPPoE session is established between a device and a radio on behalf of every other device and radio neighbor located in the MANET. These Layer 2 sessions are the means by which radio network status gets reported to the Layer 3 processes in the device. The figure below shows the PPPoE session exchange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

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



This capability requires that a Radio-Aware Routing (RAR)-compliant radio be connected to a device through Ethernet. The device always considers the Ethernet link to be up. If the radio side of the link goes down, the device waits until a routing update timeout occurs to declare the route down and then updates the routing table. The figure below shows a simple device-to-radio link topology.

Figure 2: Device-to-Radio Link



Neighbor Up and Down Signaling

Mobile Ad Hoc Networks (MANETs) are highly dynamic environments. Nodes might 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 devices. Routing protocols normally use timer-driven hello messages or neighbor timeouts to track topology changes, but MANETs reliance on these mechanisms can result in unacceptably slow convergence.

The neighbor up/down signaling capability provides faster network convergence by using link-status signals generated by the radio. The radio notifies the device each time a link to another neighbor is established or terminated by the creation and termination of PPP over Ethernet (PPPoE) sessions. In the device, the routing protocols (Open Shortest Path First version 3 [OSPFv3] or Enhanced Interior Gateway Routing Protocol [EIGRP]) respond immediately to these signals by expediting 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 device 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 virtual multipoint interfaces (VMIs) with PPPoE are used and a partner node has left or a new one has joined, the radio informs the device immediately of the topology change. Upon receiving the signal, the device immediately declares the change and updates the routing tables. The signaling capability provides these advantages:

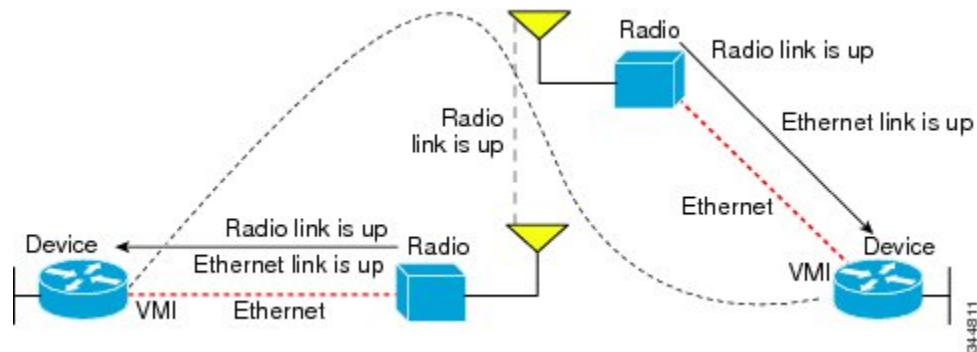
- Reduces routing delays and prevents applications from timing out.
- Enables network-based applications and information to be delivered reliably and quickly over directional radio links.
- Provides faster convergence and optimal route selection so that delay-sensitive traffic such as voice and video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

The messaging allows for flexible rerouting when necessary, in the following circumstances:

- Fading of the radio links
- Congestion of the radio links
- Radio link power fade
- Utilization of the radio

The figure below shows the signaling sequence that occurs when radio links go up and down:

Figure 3: Up and Down Signaling Sequence



PPPoE Credit-based and Metric-based Scaling and Flow Control

Each radio initiates a PPP over Ethernet (PPPoE) session with its local device 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 (device-to-device). This process is duplicated each time a radio establishes a new link.

The carrying capacity of each radio link might vary due to location changes or environmental conditions, and many radio transmission systems have limited buffering capabilities. To minimize the need for packet queueing in the radio, PPPoE protocol extensions enable the device to control traffic buffering in congestion situations. Implementing flow-control on these device-to-radio sessions allows use of quality of service (QoS) features such as fair queueing.

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 device acknowledges the request, all subsequent traffic must be flow controlled. If a flow-control session is requested and cannot be supported by the device, the session is terminated. Typically, both the radio and the device initially grant credits during session discovery. Once a device exhausts its credits, it must stop sending until additional credits are granted. Credits can be added incrementally over the course of a session.

Metrics scaling is used with high-performance radios that require high-speed links. 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 display the maximum and current data rates and the scalar value set by the radio in the **show vmi neighbor detail** command output.

System Components

The Radio Aware Routing (RAR) feature is implemented using the MANET infrastructure comprising of different components such as PPPoE, Virtual Multipoint Interface (VMI), QoS, routing protocol interface and RAR protocols.

Virtual Multipoint Interface (VMI)

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

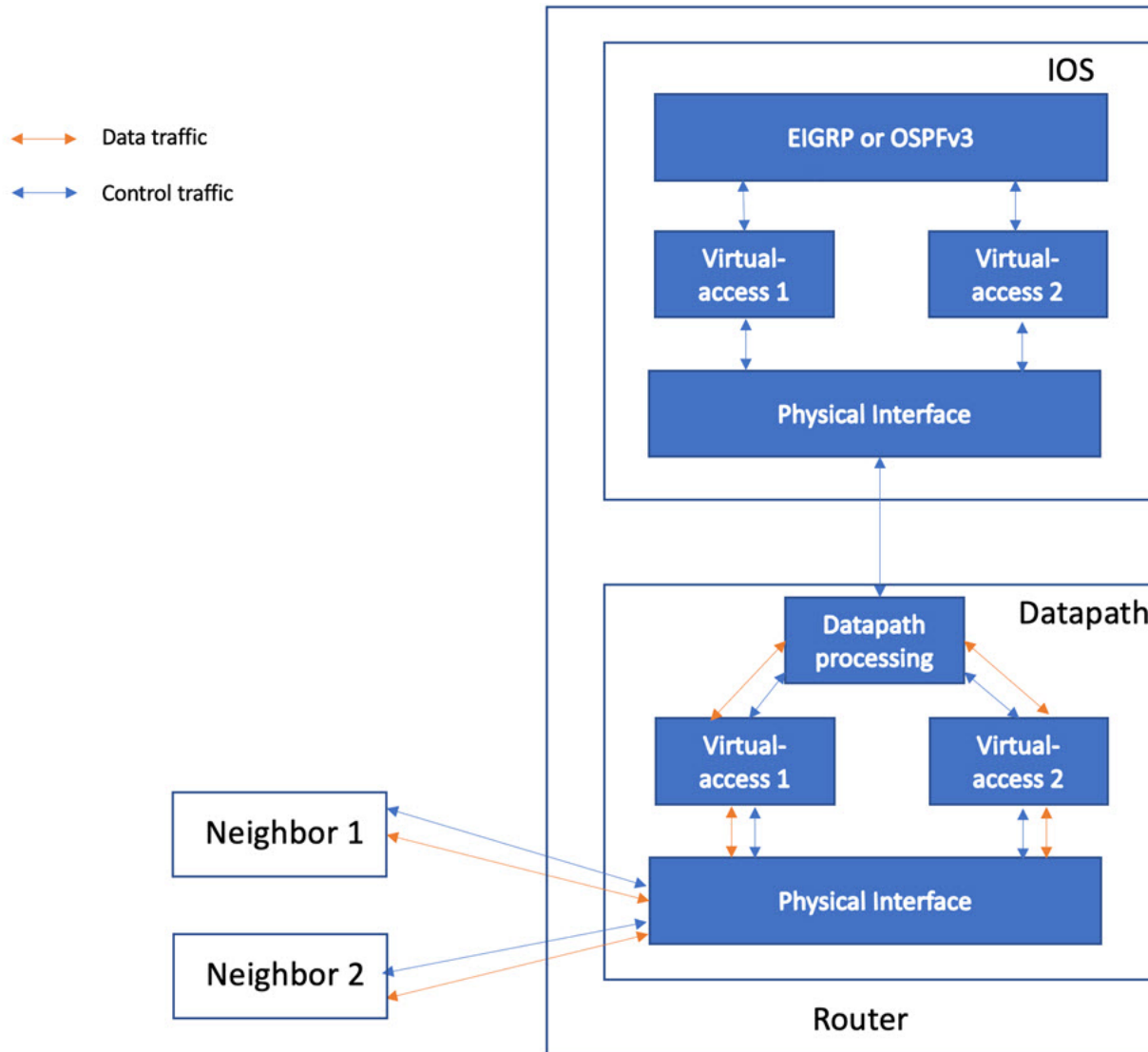
The Packet Flow diagram below illustrates the packet flow for both control and data packets over a PPPoE session when the VMI interface is in bypass mode.

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 4: Packet Flow



Restrictions

- Multicast traffic is not supported in aggregate mode.
- Cisco High Availability (HA) technology is not supported.

Enabling IPv6 Routing

Procedure

	Command or Action	Purpose
Step 1	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ipv6 unicast-routing</code> Example: Router(config)# <code>ipv6 unicast-routing</code>	Enable IPv6 Unicast routing.
Step 3	<code>ipv6 multicast-routing</code> Example: Router(config)# <code>ipv6 multicast-routing</code>	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	<code>configure terminal</code> Example: Router# <code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>subscriber templating</code> Example: Router(config)# <code>subscriber templating</code>	Configure subscriber templating.
Step 3	<code>subscriber authorization enable</code> Example: Router(config)# <code>subscriber authorization enable</code>	Enable Subscriber Service Switch type authorization.

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:	Configure QoS policy for the Virtual Template Interface.

	Command or Action	Purpose
	Router(config-pmap-c)# police 10000000 conform-action transmit exceed-action drop	

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.
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.

	Command or Action	Purpose
Step 2	interface <i>[type slot/port]</i> 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 <i>group_name</i> 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 <i>number</i> Example: Router(config)# interface Virtual-Template 1	Creates a Virtual Template Interface for configuration and dynamic application to Virtual Access Interfaces.
Step 3	mtu <i>size</i> Example: Router(config-if)# mtu 1484	Sets the MTU size.
Step 4	ip unnumbered <i>interface_type interface_number</i> 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:	Disables ICMP redirect messages.

	Command or Action	Purpose
	Router(config-if)# no ip redirects	
Step 6	ip tcp adjust-mss <i>size</i> Example: Router(config-if)# ip tcp adjust-mss 1444	Adjust the mss of transit packets.
Step 7	load-interval <i>value</i> 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 <i>size</i> Example: Router(config-if)# ipv6 mtu 1484	Sets the IPv6 MTU size.
Step 11	ospfv3 1 network <i>type</i> Example: Router(config-if)# ospfv3 1 network manet	Configure the Virtual Template Interface as MANET OSPF interface type.
Step 12	ospfv3 1 hello-interval <i>value</i> Example: Router(config-if)# ospfv3 1 hello-interval 10	Configure the hello interval value for OSPFv3.
Step 13	ospfv3 1 ipv4 area <i>id</i> 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 <i>id</i> Example: Router(config-if)# ospfv3 1 ipv6 area 0	Enable Virtual Template Interface to participate in OSPFv3 IPv6 routing.
Step 15	service-policy input <i>policy_map_name</i> Example: Router(config-if)# service-policy input rar_policer	Attach the created policy map name to the Virtual Template Interface.

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.
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 <i>process_id</i> Example: Router(config)# router ospfv3 1	Global configuration for OSPFv3.
Step 3	router-id <i>ipv4_address</i> Example: Router(config-router)# router-id 101.101.101.101	Router ID for OSPFv3.
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 <i>value</i> metric-type <i>type</i> 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 <i>process_id</i> Example:	Global configuration for OSPFv3.

	Command or Action	Purpose
	Router(config)# router ospfv3 1	
Step 3	router-id <i>ipv4_address</i> 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 <i>value</i> metric-type <i>type</i> 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.

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
PADG xmit: 127926272   rcvd: 36
In-band credit pkt xmit: 0   rcvd: 0
Last credit packet snapshot
PADG xmit: seq_num = 38, fc_n = 65535, bcn = 0

```



```

PADG rcvd: seq_num = 38, fcn = 1953, bcn = 65535
PADG rcvd: seq_num = 0, fcn = 0, bcn = 61
PADG 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
PADG 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
PADG rcvd: seq_num = 42, fcn = 1953, bcn = 65535
PADG rcvd: seq_num = 0, fcn = 0, bcn = 66
PADG 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 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 to the 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=vml,
    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
PADG 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
  PADG rcvd: seq_num = 198, fcn = 1954, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 105
  PADG 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 vml 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=vml,
    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#

```

