



Routing Configuration Guide, Cisco IOS XE Gibraltar 16.10.x (Catalyst 9200 Switches)

First Published: 2018-12-10

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

CHAPTER 1

Configuring Bidirectional Forwarding Detection 1

Bidirectional Forwarding Detection	1
Prerequisites for Bidirectional Forwarding Detection	1
Restrictions for Bidirectional Forwarding Detection	1
Information About Bidirectional Forwarding Detection	2
BFD Operation	2
Neighbor Relationships	2
BFD Detection of Failures	3
BFD Version Interoperability	3
BFD Session Limits	3
BFD Support for Nonbroadcast Media Interfaces	3
BFD Support for Nonstop Forwarding with Stateful Switchover	4
BFD Support for Stateful Switchover	4
BFD Support for Static Routing	4
Benefits of Using BFD for Failure Detection	5
How to Configure Bidirectional Forwarding Detection	5
Configuring BFD Session Parameters on the Interface	5
Configuring BFD Support for Dynamic Routing Protocols	7
Configuring BFD Support for eBGP	7
Configuring BFD Support for EIGRP	8
Configuring BFD Support for IS-IS	10
Configuring BFD Support for OSPF	14
Configuring BFD Support for HSRP	17
Configuring BFD Support for Static Routing	19
Configuring BFD Echo Mode	21
Prerequisites	22

Restrictions	22
Disabling BFD Echo Mode Without Asymmetry	22
Creating and Configuring BFD Templates	23
Configuring a Single-Hop Template	23
Monitoring and Troubleshooting BFD	24
Monitoring and Troubleshooting BFD	24
Feature Information for Bidirectional Forwarding Detection	25

CHAPTER 2**Configuring BFD-EIGRP Support 27**

BFD - EIGRP Support	27
Prerequisites for BFD-EIGRP Support	27
Information About BFD - EIGRP Support	27
Overview of BFD-EIGRP Support	27
How to Configure BFD - EIGRP Support	28
Configuring BFD - EIGRP Support	28
Configuration Examples for BFD - EIGRP Support	29
Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default	29
Feature Information for BFD-EIGRP Support	35

CHAPTER 3**Configuring IP Unicast Routing 37**

Restrictions for IP Unicast Routing	37
Information About Configuring IP Unicast Routing	37
Information About IP Routing	37
Types of Routing	38
Classless Routing	39
Address Resolution	41
Proxy ARP	42
ICMP Router Discovery Protocol	42
UDP Broadcast Packets and Protocols	42
Broadcast Packet Handling	43
IP Broadcast Flooding	43
How to Configure IP Routing	44
How to Configure IP Addressing	45
Default IP Addressing Configuration	45

Assigning IP Addresses to Network Interfaces	46
Using Subnet Zero	48
Disabling Classless Routing	49
Configuring Address Resolution Methods	49
Defining a Static ARP Cache	50
Setting ARP Encapsulation	51
Enabling Proxy ARP	52
Routing Assistance When IP Routing is Disabled	53
Proxy ARP	53
Default Gateway	53
ICMP Router Discovery Protocol (IRDP)	54
Configuring Broadcast Packet Handling	56
Enabling Directed Broadcast-to-Physical Broadcast Translation	56
Forwarding UDP Broadcast Packets and Protocols	58
Establishing an IP Broadcast Address	59
Flooding IP Broadcasts	60
Monitoring and Maintaining IP Addressing	62
How to Configure IP Unicast Routing	63
Enabling IP Unicast Routing	63
Example of Enabling IP Routing	64
What to Do Next	64
Monitoring and Maintaining the IP Network	64
Feature Information for IP Unicast Routing	64

CHAPTER 4
Configuring RIP 65

Information About RIP	65
Summary Addresses and Split Horizon	65
How to Configure RIP	66
Default RIP Configuration	66
Configuring Basic RIP Parameters	66
Configuring RIP Authentication	68
Configuring Summary Addresses and Split Horizon	70
Configuring Split Horizon	71
Configuration Example for Summary Addresses and Split Horizon	72

Feature Information for Routing Information Protocol 73

CHAPTER 5

Configuring OSPF 75

Information About OSPF 75

OSPF Nonstop Forwarding 75

OSPF NSF Awareness 76

OSPF NSF Capability 76

OSPF Area Parameters 76

Other OSPF Parameters 76

LSA Group Pacing 77

Loopback Interfaces 77

How to Configure OSPF 78

Default OSPF Configuration 78

Configuring Basic OSPF Parameters 79

Configuring OSPF Interfaces 80

Configuring OSPF Area Parameters 82

Configuring Other OSPF Parameters 84

Changing LSA Group Pacing 86

Configuring a Loopback Interface 87

Monitoring OSPF 88

Configuration Examples for OSPF 88

Example: Configuring Basic OSPF Parameters 88

Feature Information for OSPF 89

CHAPTER 6

Configuring EIGRP 91

Information About EIGRP 91

EIGRP Features 91

EIGRP Components 92

EIGRP Nonstop Forwarding 92

EIGRP NSF Awareness 93

EIGRP NSF Capability 93

EIGRP Stub Routing 93

How to Configure EIGRP 94

Default EIGRP Configuration 94

Configuring Basic EIGRP Parameters	96
Configuring EIGRP Interfaces	97
Configuring EIGRP Route Authentication	99
Monitoring and Maintaining EIGRP	101
Feature Information for EIGRP	101

CHAPTER 7
Configuring IS-IS Routing 103

Information About IS-IS Routing	103
Nonstop Forwarding Awareness	104
IS-IS Global Parameters	104
IS-IS Interface Parameters	104
How to Configure IS-IS	105
Default IS-IS Configuration	105
Enabling IS-IS Routing	106
Configuring IS-IS Global Parameters	108
Configuring IS-IS Interface Parameters	111
Monitoring and Maintaining IS-IS	113
Feature Information for IS-IS	113

CHAPTER 8
Configuring Multi-VRF CE 115

Information About Multi-VRF CE	115
Understanding Multi-VRF CE	115
Network Topology	116
Packet-Forwarding Process	117
Network Components	117
VRF-Aware Services	117
How to Configure Multi-VRF CE	118
Default Multi-VRF CE Configuration	118
Multi-VRF CE Configuration Guidelines	118
Configuring VRFs	119
Configuring VRF-Aware Services	120
Configuring VRF-Aware Services for ARP	121
Configuring VRF-Aware Services for Ping	121
Configuring VRF-Aware Services for SNMP	121

Configuring VRF-Aware Services for NTP	122
Configuring VRF-Aware Services for uRPF	125
Configuring VRF-Aware RADIUS	126
Configuring VRF-Aware Services for Syslog	126
Configuring VRF-Aware Services for Traceroute	127
Configuring VRF-Aware Services for FTP and TFTP	127
Configuring Multicast VRFs	128
Configuring a VPN Routing Session	130
Monitoring Multi-VRF CE	131
Configuration Examples for Multi-VRF CE	131
Multi-VRF CE Configuration Example	131
Feature Information for Multi-VRF CE	135

CHAPTER 9	Configuring Unicast Reverse Path Forwarding	137
	Configuring Unicast Reverse Path Forwarding	137

CHAPTER 10	Protocol-Independent Features	139
	Protocol-Independent Features	139
	Distributed Cisco Express Forwarding	139
	Information About Cisco Express Forwarding	139
	How to Configure Cisco Express Forwarding	140
	Load-Balancing Scheme for CEF Traffic	141
	Restrictions for Configuring a Load-Balancing Scheme for CEF Traffic	141
	CEF Load-Balancing Overview	141
	Per-Destination Load Balancing for CEF Traffic	142
	Load-Balancing Algorithms for CEF Traffic	142
	How to Configure a Load-Balancing for CEF Traffic	142
	Configuration Examples for CEF Traffic Load-Balancing	144
	Number of Equal-Cost Routing Paths	145
	Information About Equal-Cost Routing Paths	145
	How to Configure Equal-Cost Routing Paths	145
	Static Unicast Routes	146
	Information About Static Unicast Routes	146
	Configuring Static Unicast Routes	146

Default Routes and Networks	148
Information About Default Routes and Networks	148
How to Configure Default Routes and Networks	148
Route Maps to Redistribute Routing Information	149
Information About Route Maps	149
How to Configure a Route Map	149
How to Control Route Distribution	153
Policy-Based Routing	154
Restrictions for Configuring PBR	154
Information About Policy-Based Routing	155
How to Configure PBR	156
Filtering Routing Information	158
Setting Passive Interfaces	159
Controlling Advertising and Processing in Routing Updates	160
Filtering Sources of Routing Information	161
Managing Authentication Keys	162
Prerequisites	162
How to Configure Authentication Keys	162

CHAPTER 11
Configuring Generic Routing Encapsulation(GRE) Tunnel IP Source and Destination VRF Membership 165

Restrictions for GRE Tunnel IP Source and Destination VRF Membership	165
Information About GRE Tunnel IP Source and Destination VRF Membership	165
How to Configure GRE Tunnel IP Source and Destination VRF Membership	166
Configuration Example for GRE Tunnel IP Source and Destination VRF Membership	167
Additional References	168
Feature History for Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership	169



CHAPTER 1

Configuring Bidirectional Forwarding Detection

- [Bidirectional Forwarding Detection, on page 1](#)
- [Prerequisites for Bidirectional Forwarding Detection, on page 1](#)
- [Restrictions for Bidirectional Forwarding Detection, on page 1](#)
- [Information About Bidirectional Forwarding Detection, on page 2](#)
- [How to Configure Bidirectional Forwarding Detection, on page 5](#)
- [Feature Information for Bidirectional Forwarding Detection, on page 25](#)

Bidirectional Forwarding Detection

This document describes how to enable the Bidirectional Forwarding Detection (BFD) protocol. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.

BFD provides a consistent failure detection method for network administrators, in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and convergence time will be consistent and predictable.

Prerequisites for Bidirectional Forwarding Detection

- Cisco Express Forwarding and IP routing must be enabled on all participating switches.
- One of the IP routing protocols supported by BFD must be configured on the switches before BFD is deployed. You should implement fast convergence for the routing protocol that you are using. See the IP routing documentation for your version of Cisco IOS software for information on configuring fast convergence. See the Restrictions for Bidirectional Forwarding Detection section for more information on BFD routing protocol support in Cisco IOS software.

Restrictions for Bidirectional Forwarding Detection

- BFD works only for directly connected neighbors. BFD neighbors must be no more than one IP hop away. Multihop configurations are not supported.

- BFD support is not available for all platforms and interfaces. To confirm BFD support for a specific platform or interface and obtain the most accurate platform and hardware restrictions, see the Cisco IOS software release notes for your software version.
- BFD packets are not matched in the QoS policy for self-generated packets.
- BFD packets are matched in the **class class-default** command. So, the user must make sure of the availability of appropriate bandwidth to prevent dropping of BFD packets due to oversubscription.
- BFD HA is not supported.

Information About Bidirectional Forwarding Detection

BFD Operation

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent routers, including the interfaces, data links, and forwarding planes.

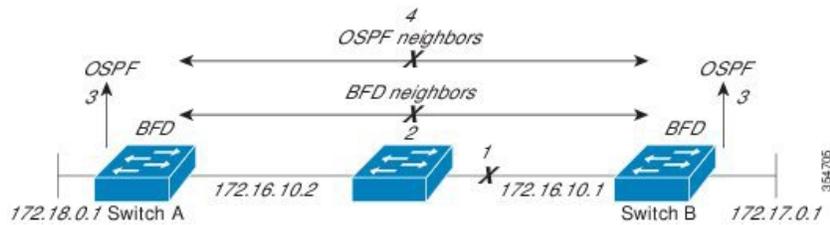
BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers. Therefore, in order for a BFD session to be created, you must configure BFD on both systems (or BFD peers). Once BFD has been enabled on the interfaces and at the router level for the appropriate routing protocols, a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

Neighbor Relationships

BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols BGP, EIGRP, IS-IS, and OSPF. By sending rapid failure detection notices to the routing protocols in the local router to initiate the routing table recalculation process, BFD contributes to greatly reduced overall network convergence time. The figure below shows a simple network with two routers running OSPF and BFD. When OSPF discovers a neighbor (1) it sends a request to the local BFD process to initiate a BFD neighbor session with the OSPF neighbor router (2). The BFD neighbor session with the OSPF neighbor router is established (3).



The figure below shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor router is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the routers will immediately start converging on it.



A routing protocol needs to register with BFD for every neighbor it acquires. Once a neighbor is registered, BFD initiates a session with the neighbor if a session does not already exist.

OSPF registers with BFD when:

- A neighbor finite state machine (FSM) transitions to full state.
- Both OSPF BFD and BFD are enabled.

On broadcast interfaces, OSPF establishes a BFD session only with the designated router (DR) and backup designated router (BDR), but not between any two routers in DROTHER state.

BFD Detection of Failures

Once a BFD session has been established and timer negotiations are complete, BFD peers send BFD control packets that act in the same manner as an IGP hello protocol to detect liveness, except at a more accelerated rate. The following information should be noted:

- BFD is a forwarding path failure detection protocol. BFD detects a failure, but the routing protocol must take action to bypass a failed peer.
- Starting Cisco IOS XE Denali 16.3.1, Cisco devices will support BFD version 0, where devices will use one BFD session for multiple client protocols in the implementation. For example, if a network is running OSPF and EIGRP across the same link to the same peer, only one BFD session will be established, and BFD will share session information with both routing protocols.

BFD Version Interoperability

All BFD sessions come up as Version 1 by default and will be interoperable with Version 0. The system automatically performs BFD version detection, and BFD sessions between neighbors will run in the highest common BFD version between neighbors. For example, if one BFD neighbor is running BFD Version 0 and the other BFD neighbor is running Version 1, the session will run BFD Version 0. The output from the **show bfd neighbors [details]** command will verify which BFD version a BFD neighbor is running.

See the Example Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default for an example of BFD version detection.

BFD Session Limits

Starting Cisco IOS XE Denali 16.3.1, the number of BFD sessions that can be created has been increased to 100.

BFD Support for Nonbroadcast Media Interfaces

Starting Cisco IOS XE Denali 16.3.1, the BFD feature is supported on routed, SVI and L3 portchannels.

The **bfd interval** command must be configured on the interface to initiate BFD monitoring.

BFD Support for Nonstop Forwarding with Stateful Switchover

Typically, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Nonstop forwarding (NSF) helps to suppress routing flaps in devices that are enabled with stateful switchover (SSO), thereby reducing network instability.

NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored after a switchover. With NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors while the standby RP assumes control from the failed active RP during a switchover. The ability of line cards and forwarding processors to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP is key to NSF operation.

In devices that support dual RPs, SSO establishes one of the RPs as the active processor; the other RP is designated as the standby processor, and then synchronizes information between them. A switchover from the active to the standby processor occurs when the active RP fails, when it is removed from the networking device, or when it is manually taken down for maintenance.

BFD Support for Stateful Switchover

The BFD protocol provides short-duration detection of failures in the path between adjacent forwarding engines. In network deployments that use dual RP routers or switches (to provide redundancy), the routers have a graceful restart mechanism that protects the forwarding state during a switchover between the active RP and the standby RP.

The dual RPs have variable switchover times that depend on the ability of the hardware to detect a communication failure. When BFD is running on the RP, some platforms are not able to detect a switchover before the BFD protocol times out; these platforms are referred to as slow switchover platforms.

BFD Support for Static Routing

Unlike dynamic routing protocols, such as OSPF and BGP, static routing has no method of peer discovery. Therefore, when BFD is configured, the reachability of the gateway is completely dependent on the state of the BFD session to the specified neighbor. Unless the BFD session is up, the gateway for the static route is considered unreachable, and therefore the affected routes will not be installed in the appropriate Routing Information Base (RIB).

For a BFD session to be successfully established, BFD must be configured on the interface on the peer and there must be a BFD client registered on the peer for the address of the BFD neighbor. When an interface is used by dynamic routing protocols, the latter requirement is usually met by configuring the routing protocol instances on each neighbor for BFD. When an interface is used exclusively for static routing, this requirement must be met by configuring static routes on the peers.

If a BFD configuration is removed from the remote peer while the BFD session is in the up state, the updated state of the BFD session is not signaled to IPv4 static. This will cause the static route to remain in the RIB. The only workaround is to remove the IPv4 static BFD neighbor configuration so that the static route no longer tracks BFD session state. Also, if you change the encapsulation type on a serial interface to one that is unsupported by BFD, BFD will be in a down state on that interface. The workaround is to shut down the interface, change to a supported encapsulation type, and then reconfigure BFD.

A single BFD session can be used by an IPv4 static client to track the reachability of next hops through a specific interface. You can assign a BFD group for a set of BFD-tracked static routes. Each group must have one active static BFD configuration, one or more passive BFD configurations, and the corresponding static

routes to be BFD-tracked. Nongroup entries are BFD-tracked static routes for which a BFD group is not assigned. A BFD group must accommodate static BFD configurations that can be part of different VRFs. Effectively, the passive static BFD configurations need not be in the same VRF as that of the active configuration.

For each BFD group, there can be only one active static BFD session. You can configure the active BFD session by adding a static BFD configuration and a corresponding static route that uses the BFD configuration. The BFD session in a group is created only when there is an active static BFD configuration and the static route that uses the static BFD configuration. When the active static BFD configuration or the active static route is removed from a BFD group, all the passive static routes are withdrawn from the RIB. Effectively, all the passive static routes are inactive until an active static BFD configuration and a static route to be tracked by the active BFD session are configured in the group.

Similarly, for each BFD group, there can be one or more passive static BFD configurations and their corresponding static routes to be BFD-tracked. Passive static session routes take effect only when the active BFD session state is reachable. Though the active BFD session state of the group is reachable, the passive static route is added to the RIB only if the corresponding interface state is up. When a passive BFD session is removed from a group, it will not affect the active BFD session if one existed, or the BFD group reachability status.

Benefits of Using BFD for Failure Detection

When you deploy any feature, it is important to consider all the alternatives and be aware of any trade-offs being made.

The closest alternative to BFD in conventional EIGRP, IS-IS, and OSPF deployments is the use of modified failure detection mechanisms for EIGRP, IS-IS, and OSPF routing protocols.

If you set EIGRP hello and hold timers to their absolute minimums, the failure detection rate for EIGRP falls to within a one- to two-second range.

If you use fast hellos for either IS-IS or OSPF, these Interior Gateway Protocol (IGP) protocols reduce their failure detection mechanisms to a minimum of one second.

There are several advantages to implementing BFD over reduced timer mechanisms for routing protocols:

- Although reducing the EIGRP, IS-IS, and OSPF timers can result in minimum detection timer of one to two seconds, BFD can provide failure detection in less than one second.
- Because BFD is not tied to any particular routing protocol, it can be used as a generic and consistent failure detection mechanism for EIGRP, IS-IS, and OSPF.
- Because some parts of BFD can be distributed to the data plane, it can be less CPU-intensive than the reduced EIGRP, IS-IS, and OSPF timers, which exist wholly at the control plane.

How to Configure Bidirectional Forwarding Detection

Configuring BFD Session Parameters on the Interface

To configure BFD on an interface, you need to set the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Perform one of the following steps:
 - **ip address** *ipv4-address mask*
 - **ipv6 address** *ipv6-address/mask*
4. **bfd interval** *milliseconds min_rx milliseconds multiplier interval-multiplier*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	Perform one of the following steps: <ul style="list-style-type: none"> • ip address <i>ipv4-address mask</i> • ipv6 address <i>ipv6-address/mask</i> Example: Configuring an IPv4 address for the interface: Device(config-if)# ip address 10.201.201.1 255.255.255.0 Configuring an IPv6 address for the interface: Device(config-if)# ipv6 address 2001:db8:1:1::1/32	Configures an IP address for the interface.
Step 4	bfd interval <i>milliseconds min_rx milliseconds multiplier interval-multiplier</i> Example: Device(config-if)# bfd interval 100 min_rx 100 multiplier 3	Enables BFD on the interface. The BFD interval configuration is removed when the subinterface on which it is configured is removed. The BFD interval configuration is not removed when: <ul style="list-style-type: none"> • an IPv4 address is removed from an interface • an IPv6 address is removed from an interface • IPv6 is disabled from an interface • an interface is shutdown

	Command or Action	Purpose
		<ul style="list-style-type: none"> • IPv4 CEF is disabled globally or locally on an interface • IPv6 CEF is disabled globally or locally on an interface
Step 5	end Example: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BFD Support for Dynamic Routing Protocols

Configuring BFD Support for eBGP

This section describes the procedure for configuring BFD support for BGP so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

Before you begin

e BGP must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.



Note Output from the **show bfd neighbors details** command shows the configured intervals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-tag*
4. **neighbor** *ip-address* **fall-over bfd**
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip bgp neighbor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-tag</i> Example: Device(config)# router bgp tag1	Specifies a BGP process and enters router configuration mode.
Step 4	neighbor <i>ip-address</i> fall-over bfd Example: Device(config-router)# neighbor 172.16.10.2 fall-over bfd	Enables BFD support for fallover.
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and returns the router to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: Device# show bfd neighbors detail	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip bgp neighbor Example: Device# show ip bgp neighbor	(Optional) Displays information about BGP and TCP connections to neighbors.

Configuring BFD Support for EIGRP

This section describes the procedure for configuring BFD support for EIGRP so that EIGRP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. There are two methods for enabling BFD support for EIGRP:

- You can enable BFD for all of the interfaces for which EIGRP is routing by using the **bfd all-interfaces** command in router configuration mode.
- You can enable BFD for a subset of the interfaces for which EIGRP is routing by using the **bfd interface *type number*** command in router configuration mode.

Before you begin

EIGRP must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.



Note Output from the **show bfd neighbors details** command shows the configured intervals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp *as-number***
4. Do one of the following:
 - **bfd all-interfaces**
 - **bfd interface *type number***
5. **end**
6. **show bfd neighbors [details]**
7. **show ip eigrp interfaces [*type number*] [*as-number*] [detail]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router eigrp <i>as-number</i> Example: Device(config)# router eigrp 123	Configures the EIGRP routing process and enters router configuration mode.
Step 4	Do one of the following: <ul style="list-style-type: none"> • bfd all-interfaces • bfd interface <i>type number</i> Example: Device(config-router)# bfd all-interfaces Example:	Enables BFD globally on all interfaces associated with the EIGRP routing process. or Enables BFD on a per-interface basis for one or more interfaces associated with the EIGRP routing process.

	Command or Action	Purpose
	Device(config-router)# bfd interface GigabitFastEthernet 1/0/1	
Step 5	end Example: Device(config-router) end	Exits router configuration mode and returns the router to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: Device# show bfd neighbors details	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip eigrp interfaces [type number] [as-number] [detail] Example: Device# show ip eigrp interfaces detail	(Optional) Displays the interfaces for which BFD support for EIGRP has been enabled.

Configuring BFD Support for IS-IS

This section describes the procedures for configuring BFD support for IS-IS so that IS-IS is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. There are two methods for enabling BFD support for IS-IS:

- You can enable BFD for all of the interfaces on which IS-IS is supporting IPv4 routing by using the **bfd all-interfaces** command in router configuration mode. You can then disable BFD for one or more of those interfaces using the **isis bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which IS-IS is routing by using the **isis bfd** command in interface configuration mode.

To configure BFD support for IS-IS, perform the steps in one of the following sections:

Prerequisites

IS-IS must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces that you want to run BFD sessions to BFD neighbors over must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.



Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

Configuring BFD Support for IS-IS for All Interfaces

To configure BFD on all IS-IS interfaces that support IPv4 routing, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **bfd all-interfaces**
5. **exit**
6. **interface** *type number*
7. **ip router isis** [*tag*]
8. **isis bfd** [*disable*]
9. **end**
10. **show bfd neighbors** [*details*]
11. **show clns interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router isis <i>area-tag</i> Example: Device(config)# router isis tag1	Specifies an IS-IS process and enters router configuration mode.
Step 4	bfd all-interfaces Example: Device(config-router)# bfd all-interfaces	Enables BFD globally on all interfaces associated with the IS-IS routing process.
Step 5	exit Example: Device(config-router)# exit	(Optional) Returns the router to global configuration mode.
Step 6	interface <i>type number</i> Example:	(Optional) Enters interface configuration mode.

	Command or Action	Purpose
	<code>Device(config)# interface fastethernet 6/0</code>	
Step 7	ip router isis [<i>tag</i>] Example: <code>Device(config-if)# ip router isis tag1</code>	(Optional) Enables support for IPv4 routing on the interface.
Step 8	isis bfd [disable] Example: <code>Device(config-if)# isis bfd</code>	(Optional) Enables or disables BFD on a per-interface basis for one or more interfaces associated with the IS-IS routing process. Note You should use the disable keyword only if you had earlier enabled BFD on all of the interfaces that IS-IS is associated with, using the bfd all-interfaces command in configuration mode.
Step 9	end Example: <code>Device(config-if)# end</code>	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 10	show bfd neighbors [details] Example: <code>Device# show bfd neighbors details</code>	(Optional) Displays information that can be used to verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 11	show clns interface Example: <code>Device# show clns interface</code>	(Optional) Displays information that can be used to verify if BFD for IS-IS has been enabled for a specific IS-IS interface that is associated.

Configuring BFD Support for IS-IS for One or More Interfaces

To configure BFD for only one or more IS-IS interfaces, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip router isis** [*tag*]
5. **isis bfd** [disable]
6. **end**
7. **show bfd neighbors** [details]
8. **show clns interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface fastethernet 6/0	Enters interface configuration mode.
Step 4	ip router isis [<i>tag</i>] Example: Device(config-if)# ip router isis tag1	Enables support for IPv4 routing on the interface.
Step 5	isis bfd [disable] Example: Device(config-if)# isis bfd	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the IS-IS routing process. <p>Note You should use the disable keyword only if you enabled BFD on all of the interfaces that IS-IS is associated with using the bfd all-interfaces command in router configuration mode.</p>
Step 6	end Example: Device(config-if)# end	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 7	show bfd neighbors [details] Example: Device# show bfd neighbors details	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 8	show clns interface Example: Device# show clns interface	(Optional) Displays information that can help verify if BFD for IS-IS has been enabled for a specific IS-IS interface that is associated.

Configuring BFD Support for OSPF

This section describes the procedures for configuring BFD support for OSPF so that OSPF is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPF globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPF:

- You can enable BFD for all of the interfaces for which OSPF is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ip ospf bfd [disable]** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPF is routing by using the **ip ospf bfd** command in interface configuration mode.

See the following sections for tasks for configuring BFD support for OSPF:

Configuring BFD Support for OSPF for All Interfaces

To configure BFD for all OSPF interfaces, perform the steps in this section.

If you do not want to configure BFD on all OSPF interfaces and would rather configure BFD support specifically for one or more interfaces, see the Configuring BFD Support for OSPF for One or More Interfaces section.

Before you begin

OSPF must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **bfd all-interfaces**
5. **exit**
6. **interface** *type number*
7. **ip ospf bfd [disable]**
8. **end**
9. **show bfd neighbors [details]**
10. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router ospf process-id Example: Device(config)# router ospf 4	Specifies an OSPF process and enters router configuration mode.
Step 4	bfd all-interfaces Example: Device(config-router)# bfd all-interfaces	Enables BFD globally on all interfaces associated with the OSPF routing process.
Step 5	exit Example: Device(config-router)# exit	(Optional) Returns the device to global configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 6	interface type number Example: Device(config)# interface fastethernet 6/0	(Optional) Enters interface configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 7	ip ospf bfd [disable] Example: Device(config-if)# ip ospf bfd disable	(Optional) Disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. Note You should use the disable keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the bfd all-interfaces command in router configuration mode.
Step 8	end Example: Device(config-if)# end	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 9	show bfd neighbors [details] Example: Device# show bfd neighbors detail	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 10	show ip ospf Example:	(Optional) Displays information that can help verify if BFD for OSPF has been enabled.

	Command or Action	Purpose
	Device# show ip ospf	

Configuring BFD Support for OSPF for One or More Interfaces

To configure BFD on one or more OSPF interfaces, perform the steps in this section.

Before you begin

OSPF must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf bfd** [*disable*]
5. **end**
6. **show bfd neighbors** [*details*]
7. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface fastethernet 6/0	Enters interface configuration mode.
Step 4	ip ospf bfd [<i>disable</i>] Example:	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process.

	Command or Action	Purpose
	Device(config-if)# ip ospf bfd	Note You should use the disable keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the bfd all-interfaces command in router configuration mode.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: Device# show bfd neighbors details	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip ospf Example: Device# show ip ospf	(Optional) Displays information that can help verify if BFD support for OSPF has been enabled.

Configuring BFD Support for HSRP

Perform this task to enable BFD support for Hot Standby Router Protocol (HSRP.) Repeat the steps in this procedure for each interface over which you want to run BFD sessions to HSRP peers.

HSRP supports BFD by default. If HSRP support for BFD has been manually disabled, you can reenabling it at the router level to enable BFD support globally for all interfaces or on a per-interface basis at the interface level.

Before you begin

- HSRP must be running on all participating routers.
- Cisco Express Forwarding must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef [distributed]**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **standby [group-number] ip [ip-address [secondary]]**
7. **standby bfd**
8. **exit**
9. **standby bfd all-interfaces**
10. **exit**

11. show standby neighbors

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip cef [distributed] Example: Device(config)# ip cef	Enables Cisco Express Forwarding or distributed Cisco Express Forwarding.
Step 4	interface type number Example: Device(config)# interface FastEthernet 6/0	Enters interface configuration mode.
Step 5	ip address ip-address mask Example: Device(config-if)# ip address 10.1.0.22 255.255.0.0	Configures an IP address for the interface.
Step 6	standby [group-number] ip [ip-address [secondary]] Example: Device(config-if)# standby 1 ip 10.0.0.11	Activates HSRP.
Step 7	standby bfd Example: Device(config-if)# standby bfd	(Optional) Enables HSRP support for BFD on the interface.
Step 8	exit Example: Device(config-if)# exit	Exits interface configuration mode.
Step 9	standby bfd all-interfaces Example:	(Optional) Enables HSRP support for BFD on all interfaces.

	Command or Action	Purpose
	Device(config)# standby bfd all-interfaces	
Step 10	exit Example: Device(config)# exit	Exits global configuration mode.
Step 11	show standby neighbors Example: Device# show standby neighbors	(Optional) Displays information about HSRP support for BFD.

Configuring BFD Support for Static Routing

Perform this task to configure BFD support for static routing. Repeat the steps in this procedure on each BFD neighbor. For more information, see the "Example: Configuring BFD Support for Static Routing" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Perform one of the following steps:
 - **ip address** *ipv4-address mask*
 - **ipv6 address** *ipv6-address/mask*
5. **bfd interval** *milliseconds* **mix_rx** *milliseconds* **multiplier** *interval-multiplier*
6. **exit**
7. **ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]]
8. **ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number [ip-address]*} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*]
9. **exit**
10. **show ip static route**
11. **show ip static route bfd**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface type number Example: <pre>Device(config)# interface serial 2/0</pre>	Configures an interface and enters interface configuration mode.
Step 4	Perform one of the following steps: <ul style="list-style-type: none"> • ip address ipv4-address mask • ipv6 address ipv6-address/mask Example: Configuring an IPv4 address for the interface: <pre>Device(config-if)# ip address 10.201.201.1 255.255.255.0</pre> Configuring an IPv6 address for the interface: <pre>Device(config-if)# ipv6 address 2001:db8:1:1::1/32</pre>	Configures an IP address for the interface.
Step 5	bfd interval milliseconds mix_rx milliseconds multiplier interval-multiplier Example: <pre>Device(config-if)# bfd interval 500 min_rx 500 multiplier 5</pre>	Enables BFD on the interface. The bfd interval configuration is removed when the subinterface on which it is configured is removed. The bfd interval configuration is not removed when: <ul style="list-style-type: none"> • an IPv4 address is removed from an interface • an IPv6 address is removed from an interface • IPv6 is disabled from an interface • an interface is shutdown • IPv4 CEF is disabled globally or locally on an interface • IPv6 CEF is disabled globally or locally on an interface
Step 6	exit Example: <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	<p>ip route static bfd <i>interface-type interface-number ip-address</i> [group group-name [passive]]</p> <p>Example:</p> <pre>Device(config)# ip route static bfd TenGigabitEthernet1/0/1 10.10.10.2 group group1 passive</pre>	<p>Specifies a static route BFD neighbor.</p> <ul style="list-style-type: none"> The <i>interface-type</i>, <i>interface-number</i>, and <i>ip-address</i> arguments are required because BFD support exists only for directly connected neighbors.
Step 8	<p>ip route [vrf vrf-name] <i>prefix mask</i> {<i>ip-address</i> <i>interface-type interface-number [ip-address]</i>} [dhcp] [<i>distance</i>] [name next-hop-name] [permanent track number] [tag tag]</p> <p>Example:</p> <pre>Device(config)# ip route 10.0.0.0 255.0.0.0</pre>	<p>Specifies a static route BFD neighbor.</p>
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	<p>Exits global configuration mode and returns to privileged EXEC mode.</p>
Step 10	<p>show ip static route</p> <p>Example:</p> <pre>Device# show ip static route</pre>	<p>(Optional) Displays static route database information.</p>
Step 11	<p>show ip static route bfd</p> <p>Example:</p> <pre>Device# show ip static route bfd</pre>	<p>(Optional) Displays information about the static BFD configuration from the configured BFD groups and nongroup entries.</p>
Step 12	<p>exit</p> <p>Example:</p> <pre>Device# exit</pre>	<p>Exits privileged EXEC mode and returns to user EXEC mode.</p>

Configuring BFD Echo Mode

BFD echo mode is enabled by default, but you can disable it such that it can run independently in each direction.

BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection--the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process; therefore, the number of BFD control packets that are sent out between two BFD neighbors is reduced. In addition, because the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket

delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.

Echo mode is described as without asymmetry when it is running on both sides (both BFD neighbors are running echo mode).

Prerequisites

BFD must be running on all participating routers.

Before using BFD echo mode, you must disable the sending of Internet Control Message Protocol (ICMP) redirect messages by entering the **no ip redirects** command, in order to avoid high CPU utilization.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Restrictions

BFD echo mode does not work in conjunction with Unicast Reverse Path Forwarding (uRPF) configuration. If BFD echo mode and uRPF configurations are enabled, then the sessions will flap.

Disabling BFD Echo Mode Without Asymmetry

The steps in this procedure show how to disable BFD echo mode without asymmetry—no echo packets will be sent by the router, and the router will not forward BFD echo packets that are received from any neighbor routers.

Repeat the steps in this procedure for each BFD router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no bfd echo**
4. **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	no bfd echo Example:	Disables BFD echo mode. <ul style="list-style-type: none"> • Use the no form to disable BFD echo mode.

	Command or Action	Purpose
	<code>Router(config)# no bfd echo</code>	
Step 4	end Example: <code>Router(config)# end</code>	Exits global configuration mode and returns to privileged EXEC mode.

Creating and Configuring BFD Templates

You can configure a single-hop template to specify a set of BFD interval values. BFD interval values specified as part of the BFD template are not specific to a single interface.



Note Configuring `bfd-template` will disable echo mode.

Configuring a Single-Hop Template

Perform this task to create a BFD single-hop template and configure BFD interval timers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <code>Device> enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <code>Device# configure terminal</code>	Enters global configuration mode.
Step 3	bfd-template single-hop <i>template-name</i> Example: <code>Device(config)# bfd-template single-hop bfdtemplatel</code>	Creates a single-hop BFD template and enters BFD configuration mode.

	Command or Action	Purpose
Step 4	interval min-tx milliseconds min-rx milliseconds multiplier multiplier-value Example: <pre>Device(bfd-config)# interval min-tx 120 min-rx 100 multiplier 3</pre>	Configures the transmit and receive intervals between BFD packets, and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.
Step 5	end Example: <pre>Device(bfd-config)# end</pre>	Exits BFD configuration mode and returns the device to privileged EXEC mode.

Monitoring and Troubleshooting BFD

This section describes how to retrieve BFD information for maintenance and troubleshooting. The commands in these tasks can be entered as needed, in any order desired.

This section contains information for monitoring and troubleshooting BFD for the following Cisco platforms:

Monitoring and Troubleshooting BFD

To monitor or troubleshoot BFD on Cisco 7600 series routers, perform one or more of the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show bfd neighbors [details]**
3. **debug bfd [packet | event]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show bfd neighbors [details] Example: <pre>Router# show bfd neighbors details</pre>	(Optional) Displays the BFD adjacency database. <ul style="list-style-type: none"> • The details keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	debug bfd [packet event] Example: <pre>Router# debug bfd packet</pre>	(Optional) Displays debugging information about BFD packets.

Feature Information for Bidirectional Forwarding Detection

Table 1: Feature Information for Bidirectional Forwarding Detection

Feature Name	Release	Feature Information
Bidirectional Forwarding Detection	Cisco IOS XE Fuji 16.9.2	This feature was introduced



CHAPTER 2

Configuring BFD-EIGRP Support

- [BFD - EIGRP Support, on page 27](#)

BFD - EIGRP Support

The BFD-EIGRP Support feature configures the Enhanced Interior Gateway Routing Protocol (EIGRP) with Bidirectional Forwarding Detection (BFD) so that EIGRP registers with BFD and receives all forwarding path detection failure messages from BFD.

Prerequisites for BFD-EIGRP Support

- Enhanced Interior Gateway Routing Protocol (EIGRP) must be running on all participating routers.
- The baseline parameters for Bidirectional Forwarding Detection (BFD) sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured using the **bfd** command.

Information About BFD - EIGRP Support

Overview of BFD-EIGRP Support

The BFD-EIGRP Support feature configures Bidirectional Forwarding Detection (BFD) feature for Enhanced Interior Gateway Routing Protocol (EIGRP) so that EIGRP registers with the BFD sessions on the routing interfaces, and receives forwarding path detection failure messages from BFD.

Use **bfd interval milliseconds min_rx milliseconds multiplier interval-multiplier** command to enable BFD on any interface. Use the **bfd all-interfaces** command in router configuration mode to enable BFD for all of the interfaces where EIGRP routing is enabled. Use the **bfd interface type number** command in router configuration mode to enable BFD for a subset of the interfaces where EIGRP routing is enabled.

How to Configure BFD - EIGRP Support

Configuring BFD - EIGRP Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp *as-number***
4. Do one of the following:
 - **bfd all-interfaces**
 - **bfd interface *type number***
5. **end**
6. **show bfd neighbors [details]**
7. **show ip eigrp interfaces [*type number*] [*as-number*] [detail]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router eigrp <i>as-number</i> Example: Device(config)# router eigrp 123	Configures the EIGRP routing process and enters router configuration mode.
Step 4	Do one of the following: <ul style="list-style-type: none"> • bfd all-interfaces • bfd interface <i>type number</i> Example: Device(config-router)# bfd all-interfaces Example: Device(config-router)# bfd interface Gigabit Ethernet 2/0/2	Enables BFD globally on all interfaces associated with the EIGRP routing process. or Enables BFD on a per-interface basis for one or more interfaces associated with the EIGRP routing process.

	Command or Action	Purpose
Step 5	end Example: <pre>Device(config-router)# end</pre>	Exits router configuration mode and returns the router to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: <pre>Device# show bfd neighbors details</pre>	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip eigrp interfaces [type number] [as-number] [detail] Example: <pre>Device# show ip eigrp interfaces detail</pre>	(Optional) Displays the interfaces for which BFD support for EIGRP has been enabled.

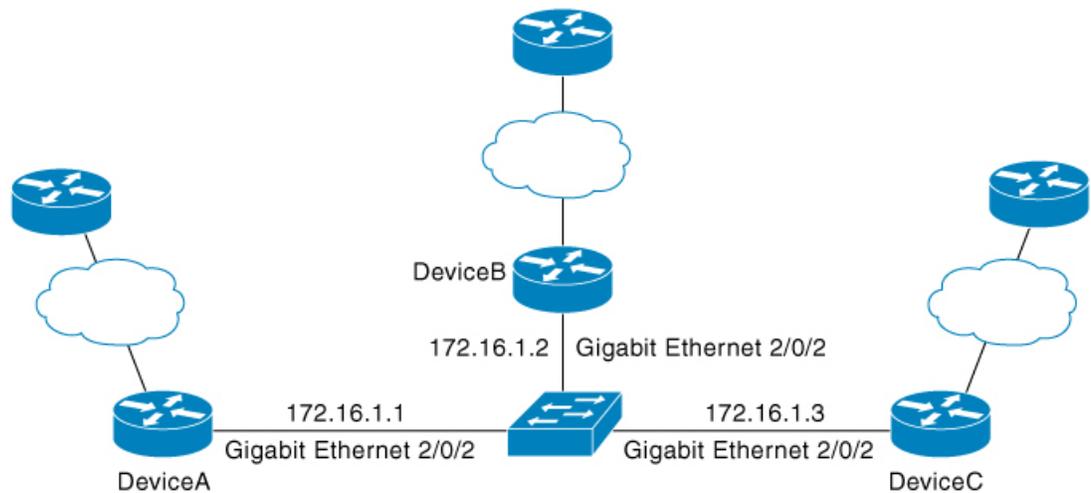
Configuration Examples for BFD - EIGRP Support

Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default

In the following example, the EIGRP network contains DeviceA, DeviceB, and DeviceC. Gigabit Ethernet 2/0/2 on DeviceA is connected to the same network as Gigabit Ethernet 2/0/2 on Device B. Gigabit Ethernet 2/0/2 on DeviceB is connected to the same network as Gigabit Ethernet 2/0/2 on DeviceC.

DeviceA and DeviceB are running BFD Version 1, which supports echo mode, and DeviceC is running BFD Version 0, which does not support echo mode. The BFD sessions between DeviceC and its BFD neighbors are said to be running echo mode with asymmetry because echo mode will run on the forwarding path for DeviceA and DeviceB, and their echo packets will return along the same path for BFD sessions and failure detections, while their BFD neighbor DeviceC runs BFD Version 0 and uses BFD control packets for BFD sessions and failure detections.

The figure below shows a large EIGRP network with several devices, three of which are BFD neighbors that are running EIGRP as their routing protocol.



3556850

The example, starting in global configuration mode, shows the configuration of BFD.

Configuration for DeviceA

```

interface Gigabit Ethernet 2/0/2
no shutdown
ip address 10.4.9.14 255.255.255.0
duplex auto
speed auto
!
interface Gigabit Ethernet 2/0/2
ip address 172.16.1.1 255.255.255.0
bfd interval 50 min_rx 50 multiplier 3
no shutdown
duplex auto
speed auto
!
router eigrp 11
network 172.16.0.0
bfd all-interfaces
auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
exec-timeout 30 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!

```

```
!  
end
```

Configuration for DeviceB

```
!  
interface Gigabit Ethernet 2/0/2  
no shutdown  
ip address 10.4.9.34 255.255.255.0  
duplex auto  
speed auto  
!  
interface Gigabit Ethernet 2/0/2  
ip address 172.16.1.2 255.255.255.0  
bfd interval 50 min_rx 50 multiplier 3  
no shutdown  
duplex auto  
speed auto  
!  
router eigrp 11  
network 172.16.0.0  
bfd all-interfaces  
auto-summary  
!  
ip default-gateway 10.4.9.1  
ip default-network 0.0.0.0  
ip route 0.0.0.0 0.0.0.0 10.4.9.1  
ip route 172.16.1.129 255.255.255.255 10.4.9.1  
!  
no ip http server  
!  
logging alarm informational  
!  
control-plane  
!  
line con 0  
exec-timeout 30 0  
stopbits 1  
line aux 0  
stopbits 1  
line vty 0 4  
login  
!  
!  
end
```

Configuration for DeviceC

```
!  
!  
interface Gigabit Ethernet 2/0/2  
no shutdown  
ip address 172.16.1.3 255.255.255.0  
duplex auto  
speed auto  
!  
interface Gigabit Ethernet 2/0/2  
ip address 172.16.1.2 255.255.255.0  
bfd interval 50 min_rx 50 multiplier 3  
no shutdown  
duplex auto
```

```

speed auto
!
router eigrp 11
network 172.16.0.0
bfd all-interfaces
auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
exec-timeout 30 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!
!
end

```

The output from the **show bfd neighbors details** command from DeviceA verifies that BFD sessions are created among all three devices and that EIGRP is registered for BFD support. The first group of output shows that DeviceC with the IP address 172.16.1.3 runs BFD Version 0 and therefore does not use the echo mode. The second group of output shows that DeviceB with the IP address 172.16.1.2 runs BFD Version 1, and the 50 millisecond BFD interval parameter had been adopted. The relevant command output is shown in bold in the output.

```
DeviceA# show bfd neighbors details
```

```

OurAddr
  NeighAddr
    LD/RD  RH/RS  Holdown(mult)  State  Int
172.16.1.1  172.16.1.3
    5/3    1(RH)    150 (3 )           Up    Fa1/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(1364284)
Rx Count: 1351813, Rx Interval (ms) min/max/avg: 28/64/49 last: 4 ms ago
Tx Count: 1364289, Tx Interval (ms) min/max/avg: 40/68/49 last: 32 ms ago
Registered protocols: EIGRP
Uptime: 18:42:45
Last packet: Version: 0
  - Diagnostic: 0
  I Hear You bit: 1      - Demand bit: 0
  Poll bit: 0           - Final bit: 0
  Multiplier: 3         - Length: 24
  My Discr.: 3          - Your Discr.: 5
  Min tx interval: 50000 - Min rx interval: 50000
  Min Echo interval: 0
OurAddr  NeighAddr

```

```

        LD/RD  RH/RS  Holdown(mult)  State      Int
172.16.1.1   172.16.1.2

        6/1    Up      0      (3 )    Up        Fal/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(317)
Rx Count: 305, Rx Interval (ms) min/max/avg: 1/1016/887 last: 448 ms ago
Tx Count: 319, Tx Interval (ms) min/max/avg: 1/1008/880 last: 532 ms ago
Registered protocols: EIGRP
Uptime: 00:04:30
Last packet: Version: 1

- Diagnostic: 0
  State bit: Up      - Demand bit: 0
  Poll bit: 0        - Final bit: 0
  Multiplier: 3      - Length: 24
  My Discr.: 1       - Your Discr.: 6
  Min tx interval: 1000000 - Min rx interval: 1000000
  Min Echo interval: 50000

```

The output from the **show bfd neighbors details** command on Device B verifies that BFD sessions have been created and that EIGRP is registered for BFD support. As previously noted, DeviceA runs BFD Version 1, therefore echo mode is running, and DeviceC runs BFD Version 0, so echo mode does not run. The relevant command output is shown in bold in the output.

DeviceB# **show bfd neighbors details**

```

OurAddr      NeighAddr
        LD/RD  RH/RS  Holdown(mult)  State      Int
172.16.1.2   172.16.1.1
        1/6    Up      0      (3 )    Up        Fal/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(337)
Rx Count: 341, Rx Interval (ms) min/max/avg: 1/1008/882 last: 364 ms ago
Tx Count: 339, Tx Interval (ms) min/max/avg: 1/1016/886 last: 632 ms ago
Registered protocols: EIGRP
Uptime: 00:05:00
Last packet: Version: 1
- Diagnostic: 0
  State bit: Up      - Demand bit: 0
  Poll bit: 0        - Final bit: 0
  Multiplier: 3      - Length: 24
  My Discr.: 6       - Your Discr.: 1
  Min tx interval: 1000000 - Min rx interval: 1000000
  Min Echo interval: 50000

OurAddr      NeighAddr
        LD/RD  RH/RS  Holdown(mult)  State      Int
172.16.1.2   172.16.1.3
        3/6    1(RH)  118  (3 )    Up        Fal/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(5735)
Rx Count: 5731, Rx Interval (ms) min/max/avg: 32/72/49 last: 32 ms ago

```

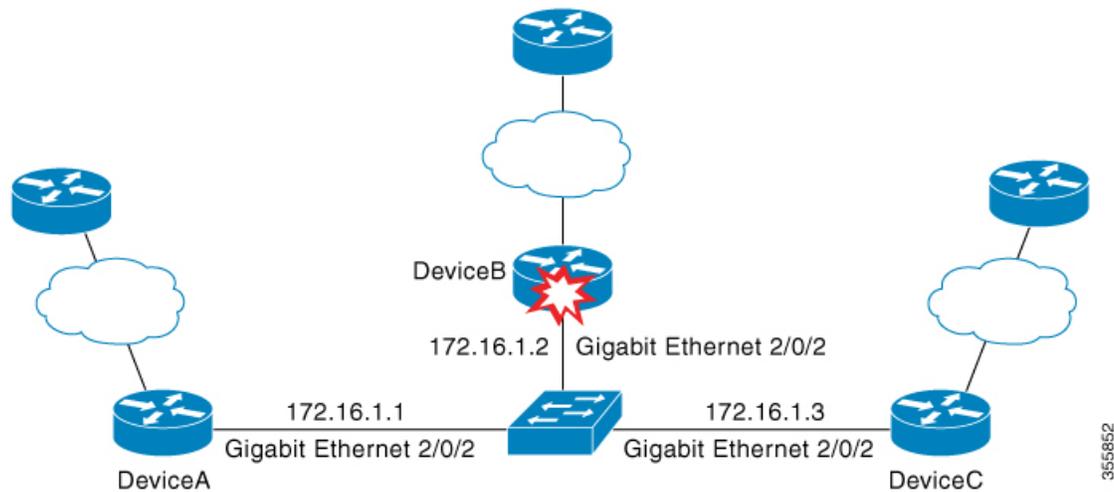
Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default

```

Tx Count: 5740, Tx Interval (ms) min/max/avg: 40/64/50 last: 44 ms ago
Registered protocols: EIGRP
Uptime: 00:04:45
Last packet: Version: 0
- Diagnostic: 0
  I Hear You bit: 1    - Demand bit: 0
  Poll bit: 0         - Final bit: 0
  Multiplier: 3       - Length: 24
  My Discr.: 6        - Your Discr.: 3
  Min tx interval: 50000 - Min rx interval: 50000
  Min Echo interval: 0

```

The figure below shows that Gigabit Ethernet 2/0/2 on DeviceB has failed. When Gigabit Ethernet 2/0/2 on DeviceB is shut down, the BFD statistics of the corresponding BFD sessions on DeviceA and DeviceB are reduced.



When Gigabit Ethernet 2/0/2 on DeviceB fails, BFD will no longer detect Device B as a BFD neighbor for DeviceA or for DeviceC. In this example, Gigabit Ethernet 2/0/2 has been administratively shut down on DeviceB.

The following output from the **show bfd neighbors** command on DeviceA now shows only one BFD neighbor for DeviceA in the EIGRP network. The relevant command output is shown in bold in the output.

```

DeviceA# show bfd neighbors
OurAddr      NeighAddr
-----
LD/RD RH/RS Holdown(mult) State Int
172.16.1.1  172.16.1.3
5/3 1(RH) 134 (3) Up Fal/0

```

The following output from the **show bfd neighbors** command on DeviceC also now shows only one BFD neighbor for DeviceC in the EIGRP network. The relevant command output is shown in bold in the output.

```

DeviceC# show bfd neighbors
OurAddr      NeighAddr
-----
LD/RD RH Holdown(mult) State Int
172.16.1.3  172.16.1.1

```

3/5 1 114 (3) Up Fa1/0

Feature Information for BFD-EIGRP Support

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 <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Table 2: Feature Information for BFD-EIGRP Support

Feature Name	Releases	Feature Information
BFD-EIGRP Support	Cisco IOS XE Gibraltar 16.10.1	The BFD-EIGRP Support feature configures the Enhanced Interior Gateway Routing Protocol (EIGRP) with Bidirectional Forwarding Detection (BFD) so that EIGRP registers with BFD and receives all forwarding path detection failure messages from BFD. The feature was introduced.



CHAPTER 3

Configuring IP Unicast Routing

- [Restrictions for IP Unicast Routing, on page 37](#)
- [Information About Configuring IP Unicast Routing, on page 37](#)
- [Information About IP Routing, on page 37](#)
- [How to Configure IP Routing, on page 44](#)
- [How to Configure IP Addressing, on page 45](#)
- [Monitoring and Maintaining IP Addressing, on page 62](#)
- [How to Configure IP Unicast Routing, on page 63](#)
- [Monitoring and Maintaining the IP Network, on page 64](#)
- [Feature Information for IP Unicast Routing, on page 64](#)

Restrictions for IP Unicast Routing

- The switch does not support tunnel interfaces for unicast routed traffic.
- Subnetwork Access Protocol (SNAP) address resolution is not supported on this device.

Information About Configuring IP Unicast Routing

This module describes how to configure IP Version 4 (IPv4) unicast routing on the switch.

A switch stack operates and appears as a single router to the rest of the routers in the network. Basic routing functions like static routing are available with .



Note In addition to IPv4 traffic, you can also enable IP Version 6 (IPv6) unicast routing and configure interfaces to forward IPv6 traffic .

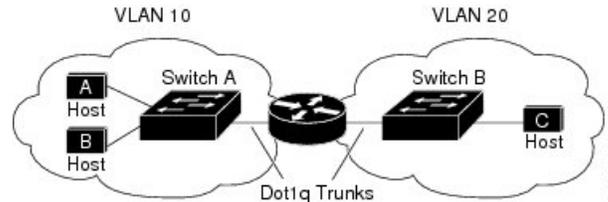
Information About IP Routing

In some network environments, VLANs are associated with individual networks or subnetworks. In an IP network, each subnetwork is mapped to an individual VLAN. Configuring VLANs helps control the size of

the broadcast domain and keeps local traffic local. However, network devices in different VLANs cannot communicate with one another without a Layer 3 device (router) to route traffic between the VLAN, referred to as inter-VLAN routing. You configure one or more routers to route traffic to the appropriate destination VLAN.

Figure 1: Routing Topology Example

This figure shows a basic routing topology. Switch A is in VLAN 10, and Switch B is in VLAN 20. The router



has an interface in each VLAN.

When Host A in VLAN 10 needs to communicate with Host B in VLAN 10, it sends a packet addressed to that host. Switch A forwards the packet directly to Host B, without sending it to the router.

When Host A sends a packet to Host C in VLAN 20, Switch A forwards the packet to the router, which receives the traffic on the VLAN 10 interface. The router checks the routing table, finds the correct outgoing interface, and forwards the packet on the VLAN 20 interface to Switch B. Switch B receives the packet and forwards it to Host C.

Types of Routing

Routers and Layer 3 switches can route packets in these ways:

- By using default routing
- By using preprogrammed static routes for the traffic
- By dynamically calculating routes by using a routing protocol

Default routing refers to sending traffic with a destination unknown to the router to a default outlet or destination.

Static unicast routing forwards packets from predetermined ports through a single path into and out of a network. Static routing is secure and uses little bandwidth, but does not automatically respond to changes in the network, such as link failures, and therefore, might result in unreachable destinations. As networks grow, static routing becomes a labor-intensive liability.

Dynamic routing protocols are used by routers to dynamically calculate the best route for forwarding traffic. There are two types of dynamic routing protocols:

- Routers using distance-vector protocols maintain routing tables with distance values of networked resources, and periodically pass these tables to their neighbors. Distance-vector protocols use one or a series of metrics for calculating the best routes. These protocols are easy to configure and use.
- Routers using link-state protocols maintain a complex database of network topology, based on the exchange of link-state advertisements (LSAs) between routers. LSAs are triggered by an event in the network, which speeds up the convergence time or time required to respond to these changes. Link-state protocols respond quickly to topology changes, but require greater bandwidth and more resources than distance-vector protocols.

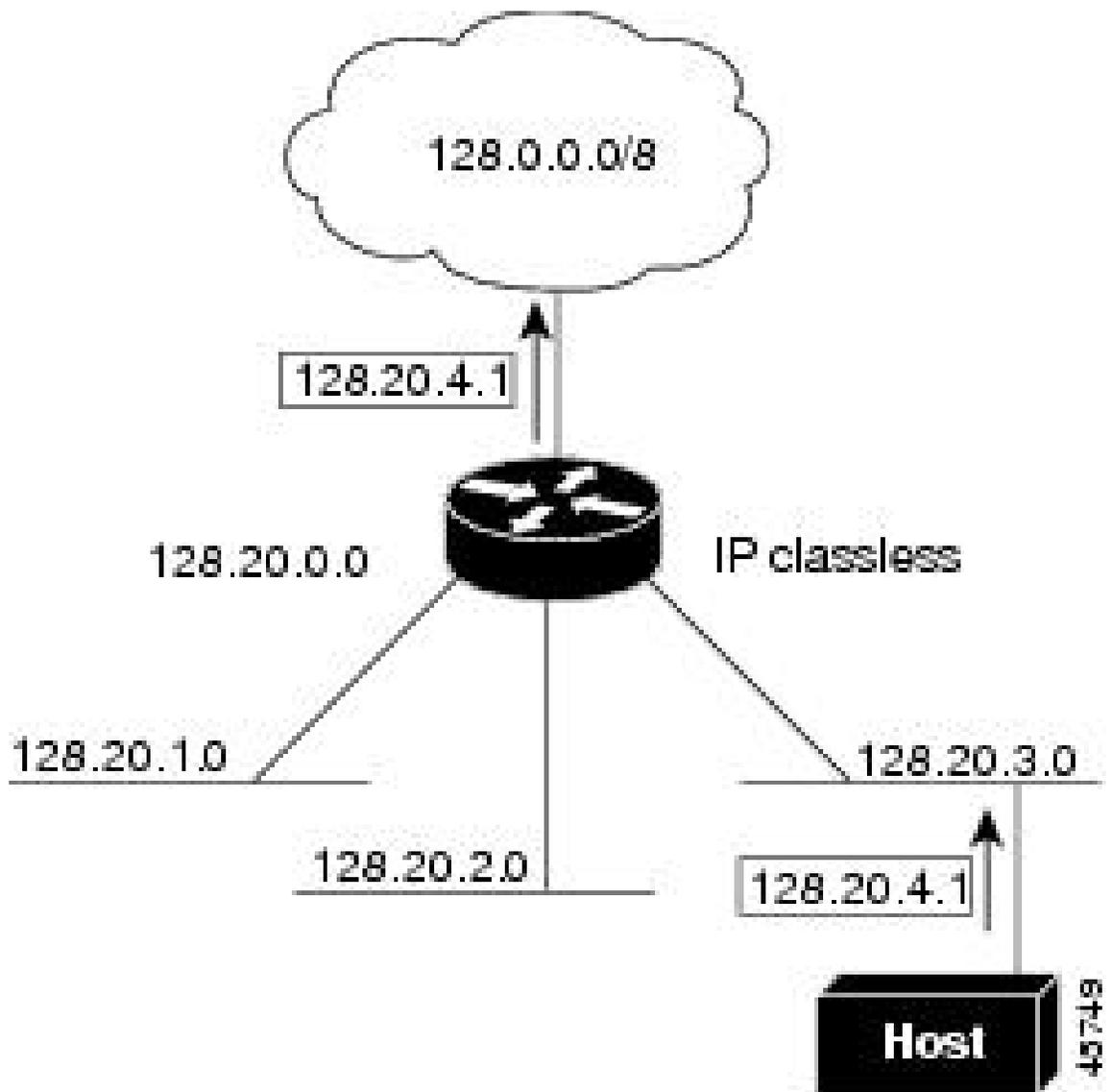
Distance-vector protocols supported by the switch are Routing Information Protocol (RIP), which uses a single distance metric (cost) to determine the best path. The switch also supports the Open Shortest Path First (OSPF) link-state protocol and Enhanced IGRP (EIGRP), which adds some link-state routing features to traditional Interior Gateway Routing Protocol (IGRP) to improve efficiency.

Classless Routing

By default, classless routing behavior is enabled on the Device when it is configured to route. With classless routing, if a router receives packets for a subnet of a network with no default route, the router forwards the packet to the best supernet route. A supernet consists of contiguous blocks of Class C address spaces used to simulate a single, larger address space and is designed to relieve the pressure on the rapidly depleting Class B address space.

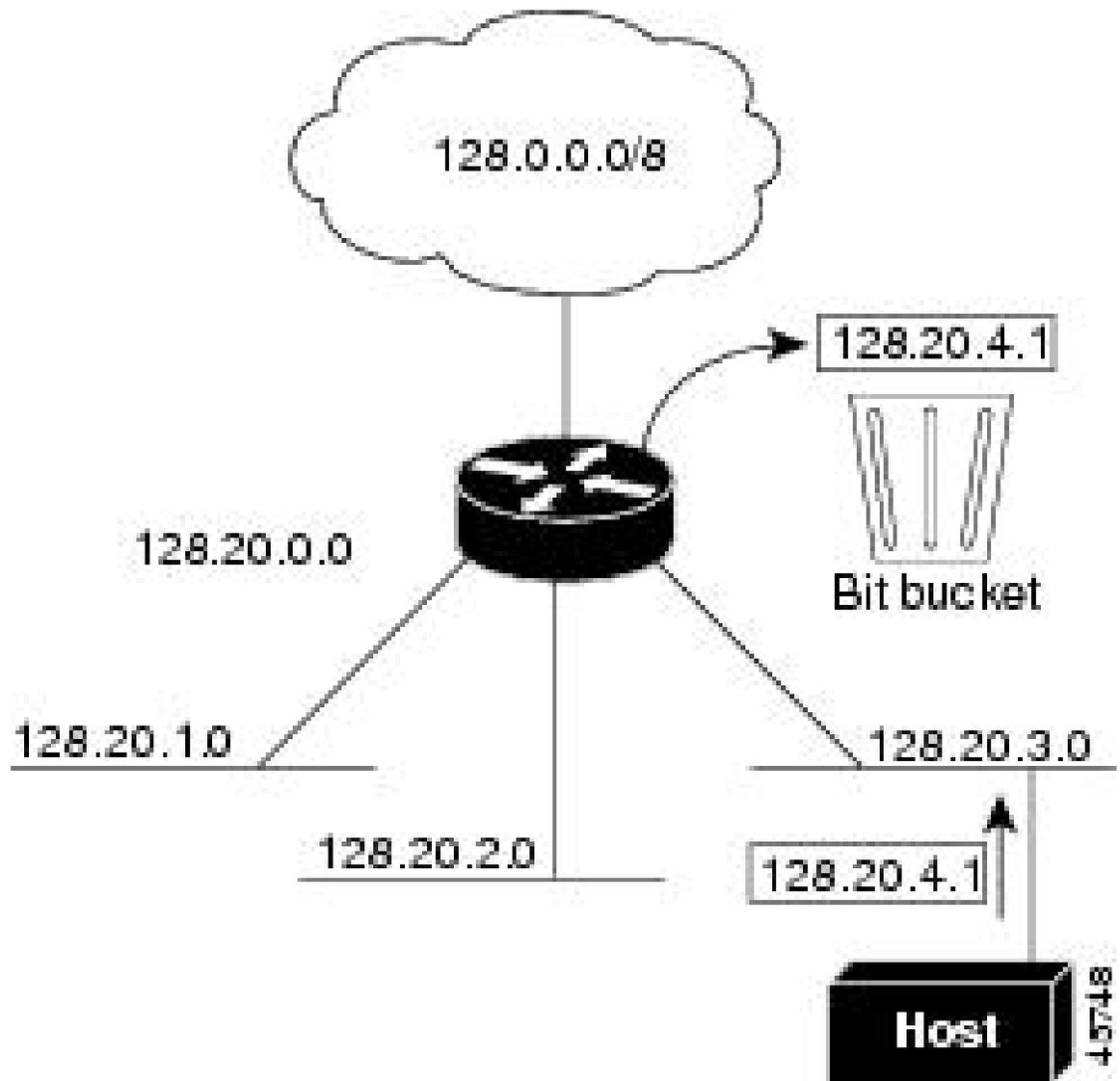
In the figure, classless routing is enabled. When the host sends a packet to 120.20.4.1, instead of discarding the packet, the router forwards it to the best supernet route. If you disable classless routing and a router receives packets destined for a subnet of a network with no network default route, the router discards the packet.

Figure 2: IP Classless Routing



In the figure, the router in network 128.20.0.0 is connected to subnets 128.20.1.0, 128.20.2.0, and 128.20.3.0. If the host sends a packet to 120.20.4.1, because there is no network default route, the router discards the packet.

Figure 3: No IP Classless Routing



To prevent the Device from forwarding packets destined for unrecognized subnets to the best supernet route possible, you can disable classless routing behavior.

Address Resolution

You can control interface-specific handling of IP by using address resolution. A device using IP can have both a local address or MAC address, which uniquely defines the device on its local segment or LAN, and a network address, which identifies the network to which the device belongs.

The local address or MAC address is known as a data link address because it is contained in the data link layer (Layer 2) section of the packet header and is read by data link (Layer 2) devices. To communicate with a device on Ethernet, the software must learn the MAC address of the device. The process of learning the MAC address from an IP address is called *address resolution*. The process of learning the IP address from the MAC address is called *reverse address resolution*.

The Device can use these forms of address resolution:

- Address Resolution Protocol (ARP) is used to associate IP address with MAC addresses. Taking an IP address as input, ARP learns the associated MAC address and then stores the IP address/MAC address association in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network.
- Proxy ARP helps hosts with no routing tables learn the MAC addresses of hosts on other networks or subnets. If the Device (router) receives an ARP request for a host that is not on the same interface as the ARP request sender, and if the router has all of its routes to the host through other interfaces, it generates a proxy ARP packet giving its own local data link address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host.

The Device also uses the Reverse Address Resolution Protocol (RARP), which functions the same as ARP does, except that the RARP packets request an IP address instead of a local MAC address. Using RARP requires a RARP server on the same network segment as the router interface. Use the **ip rarp-server address** interface configuration command to identify the server.

Proxy ARP

Proxy ARP, the most common method for learning about other routes, enables an Ethernet host with no routing information to communicate with hosts on other networks or subnets. The host assumes that all hosts are on the same local Ethernet and that they can use ARP to learn their MAC addresses. If a Device receives an ARP request for a host that is not on the same network as the sender, the Device evaluates whether it has the best route to that host. If it does, it sends an ARP reply packet with its own Ethernet MAC address, and the host that sent the request sends the packet to the Device, which forwards it to the intended host. Proxy ARP treats all networks as if they are local, and performs ARP requests for every IP address.

ICMP Router Discovery Protocol

Router discovery allows the Device to dynamically learn about routes to other networks using ICMP router discovery protocol (IRDP). IRDP allows hosts to locate routers. When operating as a client, the Device generates router discovery packets. When operating as a host, the Device receives router discovery packets. The Device can also listen to Routing Information Protocol (RIP) routing updates and use this information to infer locations of routers. The Device does not actually store the routing tables sent by routing devices; it merely keeps track of which systems are sending the data. The advantage of using IRDP is that it allows each router to specify both a priority and the time after which a device is assumed to be down if no further packets are received.

Each device discovered becomes a candidate for the default router, and a new highest-priority router is selected when a higher priority router is discovered, when the current default router is declared down, or when a TCP connection is about to time out because of excessive retransmissions.

UDP Broadcast Packets and Protocols

User Datagram Protocol (UDP) is an IP host-to-host layer protocol, as is TCP. UDP provides a low-overhead, connectionless session between two end systems and does not provide for acknowledgment of received datagrams. Network hosts occasionally use UDP broadcasts to find address, configuration, and name information. If such a host is on a network segment that does not include a server, UDP broadcasts are normally not forwarded. You can remedy this situation by configuring an interface on a router to forward certain classes of broadcasts to a helper address. You can use more than one helper address per interface.

You can specify a UDP destination port to control which UDP services are forwarded. You can specify multiple UDP protocols. You can also specify the Network Disk (ND) protocol, which is used by older diskless Sun workstations and the network security protocol SDNS.

By default, both UDP and ND forwarding are enabled if a helper address has been defined for an interface.

Broadcast Packet Handling

After configuring an IP interface address, you can enable routing and configure one or more routing protocols, or you can configure the way the Device responds to network broadcasts. A broadcast is a data packet destined for all hosts on a physical network. The Device supports two kinds of broadcasting:

- A directed broadcast packet is sent to a specific network or series of networks. A directed broadcast address includes the network or subnet fields.
- A flooded broadcast packet is sent to every network.



Note You can also limit broadcast, unicast, and multicast traffic on Layer 2 interfaces by using the **storm-control** interface configuration command to set traffic suppression levels.

Routers provide some protection from broadcast storms by limiting their extent to the local cable. Bridges (including intelligent bridges), because they are Layer 2 devices, forward broadcasts to all network segments, thus propagating broadcast storms. The best solution to the broadcast storm problem is to use a single broadcast address scheme on a network. In most modern IP implementations, you can set the address to be used as the broadcast address. Many implementations, including the one in the Device, support several addressing schemes for forwarding broadcast messages.

IP Broadcast Flooding

You can allow IP broadcasts to be flooded throughout your internetwork in a controlled fashion by using the database created by the bridging STP. Using this feature also prevents loops. To support this capability, bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured on an interface, it still can receive broadcasts. However, the interface never forwards broadcasts it receives, and the router never uses that interface to send broadcasts received on a different interface.

Packets that are forwarded to a single network address using the IP helper-address mechanism can be flooded. Only one copy of the packet is sent on each network segment.

To be considered for flooding, packets must meet these criteria. (Note that these are the same conditions used to consider packet forwarding using IP helper addresses.)

- The packet must be a MAC-level broadcast.
- The packet must be an IP-level broadcast.
- The packet must be a TFTP, DNS, Time, NetBIOS, ND, or BOOTP packet, or a UDP specified by the **ip forward-protocol udp** global configuration command.
- The time-to-live (TTL) value of the packet must be at least two.

A flooded UDP datagram is given the destination address specified with the **ip broadcast-address** interface configuration command on the output interface. The destination address can be set to any address. Thus, the destination address might change as the datagram propagates through the network. The source address is never changed. The TTL value is decremented.

When a flooded UDP datagram is sent out an interface (and the destination address possibly changed), the datagram is handed to the normal IP output routines and is, therefore, subject to access lists, if they are present on the output interface.

In the Device, the majority of packets are forwarded in hardware; most packets do not go through the Device CPU. For those packets that do go to the CPU, you can speed up spanning tree-based UDP flooding by a factor of about four to five times by using turbo-flooding. This feature is supported over Ethernet interfaces configured for ARP encapsulation.

How to Configure IP Routing

By default, IP routing is disabled on the Device, and you must enable it before routing can take place.

In the following procedures, the specified interface must be one of these Layer 3 interfaces:

- A routed port: a physical port configured as a Layer 3 port by using the **no switchport** interface configuration command.
- A switch virtual interface (SVI): a VLAN interface created by using the **interface vlan** *vlan_id* global configuration command and by default a Layer 3 interface.
- An EtherChannel port channel in Layer 3 mode: a port-channel logical interface created by using the **interface port-channel** *port-channel-number* global configuration command and binding the Ethernet interface into the channel group.

All Layer 3 interfaces on which routing will occur must have IP addresses assigned to them.



Note A Layer 3 switch can have an IP address assigned to each routed port and SVI.

Configuring routing consists of several main procedures:

- To support VLAN interfaces, create and configure VLANs on the Device or switch stack, and assign VLAN membership to Layer 2 interfaces. For more information, see the "Configuring VLANs" chapter.
- Configure Layer 3 interfaces.
- Enable IP routing on the switch.
- Assign IP addresses to the Layer 3 interfaces.
- Enable selected routing protocols on the switch.
- Configure routing protocol parameters (optional).

How to Configure IP Addressing

A required task for configuring IP routing is to assign IP addresses to Layer 3 network interfaces to enable the interfaces and allow communication with the hosts on those interfaces that use IP. The following sections describe how to configure various IP addressing features. Assigning IP addresses to the interface is required; the other procedures are optional.

- Default Addressing Configuration
- Assigning IP Addresses to Network Interfaces
- Configuring Address Resolution Methods
- Routing Assistance When IP Routing is Disabled
- Configuring Broadcast Packet Handling
- Monitoring and Maintaining IP Addressing

Default IP Addressing Configuration

Table 3: Default Addressing Configuration

Feature	Default Setting
IP address	None defined.
ARP	No permanent entries in the Address Resolution Protocol (ARP) cache. Encapsulation: Standard Ethernet-style ARP. Timeout: 14400 seconds (4 hours).
IP broadcast address	255.255.255.255 (all ones).
IP classless routing	Enabled.
IP default gateway	Disabled.
IP directed broadcast	Disabled (all IP directed broadcasts are dropped).
IP domain	Domain list: No domain names defined. Domain lookup: Enabled. Domain name: Enabled.
IP forward-protocol	If a helper address is defined or User Datagram Protocol (UDP) flooding is configured, UDP is enabled on default ports. Any-local-broadcast: Disabled. Spanning Tree Protocol (STP): Disabled. Turbo-flood: Disabled.

Feature	Default Setting
IP helper address	Disabled.
IP host	Disabled.
IRDP	Disabled. Defaults when enabled: <ul style="list-style-type: none"> • Broadcast IRDP advertisements. • Maximum interval between advertisements: 600 seconds. • Minimum interval between advertisements: 0.75 times max interval • Preference: 0.
IP proxy ARP	Enabled.
IP routing	Disabled.
IP subnet-zero	Disabled.

Assigning IP Addresses to Network Interfaces

An IP address identifies a location to which IP packets can be sent. Some IP addresses are reserved for special uses and cannot be used for host, subnet, or network addresses. RFC 1166, “Internet Numbers,” contains the official description of IP addresses.

An interface can have one primary IP address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is referred to as a subnet mask. To receive an assigned network number, contact your Internet service provider.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example:	Enters interface configuration mode, and specifies the Layer 3 interface to configure.

	Command or Action	Purpose
	Device(config)# interface gigabitethernet 1/0/1	
Step 4	no switchport Example: Device(config-if)# no switchport	Removes the interface from Layer 2 configuration mode (if it is a physical interface).
Step 5	ip address <i>ip-address subnet-mask</i> Example: Device(config-if)# ip address 10.1.5.1 255.255.255.0	Configures the IP address and IP subnet mask.
Step 6	no shutdown Example: Device(config-if)# no shutdown	Enables the physical interface.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show ip route Example: Device# show ip route	Verifies your entries.
Step 9	show ip interface [<i>interface-id</i>] Example: Device# show ip interface gigabitethernet 1/0/1	Verifies your entries.
Step 10	show running-config Example: Device# show running-config	Verifies your entries.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Using Subnet Zero

Subnetting with a subnet address of zero is strongly discouraged because of the problems that can arise if a network and a subnet have the same addresses. For example, if network 131.108.0.0 is subnetted as 255.255.255.0, subnet zero would be written as 131.108.0.0, which is the same as the network address.

You can use the all ones subnet (131.108.255.0) and even though it is discouraged, you can enable the use of subnet zero if you need the entire subnet space for your IP address.

Use the **no ip subnet-zero** global configuration command to restore the default and disable the use of subnet zero.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip subnet-zero Example: Device(config)# ip subnet-zero	Enables the use of subnet zero for interface addresses and routing updates.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show running-config Example: Device# show running-config	Verifies your entries.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Disabling Classless Routing

To prevent the Device from forwarding packets destined for unrecognized subnets to the best supernet route possible, you can disable classless routing behavior.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	no ip classless Example: Device(config)# no ip classless	Disables classless routing behavior.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show running-config Example: Device# show running-config	Verifies your entries.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Address Resolution Methods

You can perform the following tasks to configure address resolution.

Defining a Static ARP Cache

ARP and other address resolution protocols provide dynamic mapping between IP addresses and MAC addresses. Because most hosts support dynamic address resolution, you usually do not need to specify static ARP cache entries. If you must define a static ARP cache entry, you can do so globally, which installs a permanent entry in the ARP cache that the Device uses to translate IP addresses into MAC addresses. Optionally, you can also specify that the Device respond to ARP requests as if it were the owner of the specified IP address. If you do not want the ARP entry to be permanent, you can specify a timeout period for the ARP entry.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	arp ip-address hardware-address type Example: Device(config)# ip 10.1.5.1 c2f3.220a.12f4 arpa	Associates an IP address with a MAC (hardware) address in the ARP cache, and specifies encapsulation type as one of these: <ul style="list-style-type: none"> • arpa—ARP encapsulation for Ethernet interfaces • sap—HP's ARP type
Step 4	arp ip-address hardware-address type [alias] Example: Device(config)# ip 10.1.5.3 d7f3.220d.12f5 arpa alias	(Optional) Specifies that the switch respond to ARP requests as if it were the owner of the specified IP address.
Step 5	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to configure.
Step 6	arp timeout seconds Example: Device(config-if)# arp timeout 20000	(Optional) Sets the length of time an ARP cache entry will stay in the cache. The recommended value of ARP timeout is 4 hours which is also the default setting. However, if your network experiences regular updates to ARP cache entries, consider reducing the timeout. Be aware that decreasing the ARP timeout can result in increased network traffic. Setting the ARP timeout to 60 seconds or less is

	Command or Action	Purpose
		generally not recommended as it may cause network disruptions.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show interfaces [<i>interface-id</i>] Example: Device# show interfaces gigabitethernet 1/0/1	Verifies the type of ARP and the timeout value used on all interfaces or a specific interface.
Step 9	show arp Example: Device# show arp	Views the contents of the ARP cache.
Step 10	show ip arp Example: Device# show ip arp	Views the contents of the ARP cache.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Setting ARP Encapsulation

By default, Ethernet ARP encapsulation (represented by the **arpa** keyword) is enabled on an IP interface.

To disable an encapsulation type, use the **no arp arpa** interface configuration command.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	<code>interface interface-id</code> Example: Device(config)# <code>interface gigabitethernet 1/0/2</code>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	<code>end</code> Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 5	<code>show interfaces [interface-id]</code> Example: Device# <code>show interfaces</code>	Verifies ARP encapsulation configuration on all interfaces or the specified interface.
Step 6	<code>copy running-config startup-config</code> Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Enabling Proxy ARP

By default, the Device uses proxy ARP to help hosts learn MAC addresses of hosts on other networks or subnets.

Procedure

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Device> <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: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>interface interface-id</code> Example:	Enters interface configuration mode, and specifies the Layer 3 interface to configure.

	Command or Action	Purpose
	<code>Device(config)# interface gigabitethernet 1/0/2</code>	
Step 4	ip proxy-arp Example: <code>Device(config-if)# ip proxy-arp</code>	Enables proxy ARP on the interface.
Step 5	end Example: <code>Device(config)# end</code>	Returns to privileged EXEC mode.
Step 6	show ip interface [interface-id] Example: <code>Device# show ip interface gigabitethernet 1/0/2</code>	Verifies the configuration on the interface or all interfaces.
Step 7	copy running-config startup-config Example: <code>Device# copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Routing Assistance When IP Routing is Disabled

These mechanisms allow the Device to learn about routes to other networks when it does not have IP routing enabled:

- Proxy ARP
- Default Gateway
- ICMP Router Discovery Protocol (IRDP)

Proxy ARP

Proxy ARP is enabled by default. To enable it after it has been disabled, see the “Enabling Proxy ARP” section. Proxy ARP works as long as other routers support it.

Default Gateway

Another method for locating routes is to define a default router or default gateway. All non-local packets are sent to this router, which either routes them appropriately or sends an IP Control Message Protocol (ICMP) redirect message back, defining which local router the host should use. The Device caches the redirect messages and forwards each packet as efficiently as possible. A limitation of this method is that there is no means of detecting when the default router has gone down or is unavailable.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip default-gateway ip-address Example: Device(config)# ip default gateway 10.1.5.1	Sets up a default gateway (router).
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show ip redirects Example: Device# show ip redirects	Displays the address of the default gateway router to verify the setting.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

ICMP Router Discovery Protocol (IRDP)

The only required task for IRDP routing on an interface is to enable IRDP processing on that interface. When enabled, the default parameters apply.

You can optionally change any of these parameters. If you change the **maxadvertinterval** value, the **holdtime** and **minadvertinterval** values also change, so it is important to first change the **maxadvertinterval** value, before manually changing either the **holdtime** or **minadvertinterval** values.

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device> enable</pre>	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	<p>interface <i>interface-id</i></p> <p>Example:</p> <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	<p>ip irdp</p> <p>Example:</p> <pre>Device(config-if)# ip irdp</pre>	Enables IRDP processing on the interface.
Step 5	<p>ip irdp multicast</p> <p>Example:</p> <pre>Device(config-if)# ip irdp multicast</pre>	<p>(Optional) Sends IRDP advertisements to the multicast address (224.0.0.1) instead of IP broadcasts.</p> <p>Note This command allows for compatibility with Sun Microsystems Solaris, which requires IRDP packets to be sent out as multicasts. Many implementations cannot receive these multicasts; ensure end-host ability before using this command.</p>
Step 6	<p>ip irdp holdtime <i>seconds</i></p> <p>Example:</p> <pre>Device(config-if)# ip irdp holdtime 1000</pre>	(Optional) Sets the IRDP period for which advertisements are valid. The default is three times the maxadvertinterval value. It must be greater than maxadvertinterval and cannot be greater than 9000 seconds. If you change the maxadvertinterval value, this value also changes.
Step 7	<p>ip irdp maxadvertinterval <i>seconds</i></p> <p>Example:</p> <pre>Device(config-if)# ip irdp maxadvertinterval 650</pre>	(Optional) Sets the IRDP maximum interval between advertisements. The default is 600 seconds.
Step 8	<p>ip irdp minadvertinterval <i>seconds</i></p> <p>Example:</p> <pre>Device(config-if)# ip irdp minadvertinterval 500</pre>	(Optional) Sets the IRDP minimum interval between advertisements. The default is 0.75 times the maxadvertinterval . If you change the maxadvertinterval , this value changes to the new default (0.75 of maxadvertinterval).

	Command or Action	Purpose
Step 9	ip irdp preference <i>number</i> Example: Device(config-if)# ip irdp preference 2	(Optional) Sets a device IRDP preference level. The allowed range is –231 to 231. The default is 0. A higher value increases the router preference level.
Step 10	ip irdp address <i>address</i> [<i>number</i>] Example: Device(config-if)# ip irdp address 10.1.10.10	(Optional) Specifies an IRDP address and preference to proxy-advertise.
Step 11	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 12	show ip irdp Example: Device# show ip irdp	Verifies settings by displaying IRDP values.
Step 13	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Broadcast Packet Handling

Perform the tasks in these sections to enable these schemes:

- Enabling Directed Broadcast-to-Physical Broadcast Translation
- Forwarding UDP Broadcast Packets and Protocols
- Establishing an IP Broadcast Address
- Flooding IP Broadcasts

Enabling Directed Broadcast-to-Physical Broadcast Translation

By default, IP directed broadcasts are dropped; they are not forwarded. Dropping IP-directed broadcasts makes routers less susceptible to denial-of-service attacks.

You can enable forwarding of IP-directed broadcasts on an interface where the broadcast becomes a physical (MAC-layer) broadcast. Only those protocols configured by using the **ip forward-protocol** global configuration command are forwarded.

You can specify an access list to control which broadcasts are forwarded. When an access list is specified, only those IP packets permitted by the access list are eligible to be translated from directed broadcasts to

physical broadcasts. For more information on access lists, see the “Configuring ACLs” chapter in the Security section.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip directed-broadcast [<i>access-list-number</i>] Example: Device(config-if)# ip directed-broadcast 103	Enables directed broadcast-to-physical broadcast translation on the interface. You can include an access list to control which broadcasts are forwarded. When an access list, only IP packets permitted by the access list can be translated.
Step 5	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 6	ip forward-protocol {udp [<i>port</i>] nd sdns} Example: Device(config)# ip forward-protocol nd	Specifies which protocols and ports the router forwards when forwarding broadcast packets. <ul style="list-style-type: none"> • udp—Forward UPD datagrams. port: (Optional) Destination port that controls which UDP services are forwarded. • nd—Forward ND datagrams. • sdns—Forward SDNS datagrams
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	show ip interface <i>[interface-id]</i> Example: Device# show ip interface	Verifies the configuration on the interface or all interfaces
Step 9	show running-config Example: Device# show running-config	Verifies your entries.
Step 10	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Forwarding UDP Broadcast Packets and Protocols

If you do not specify any UDP ports when you configure the forwarding of UDP broadcasts, you are configuring the router to act as a BOOTP forwarding agent. BOOTP packets carry DHCP information.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip helper-address <i>address</i> Example: Device(config-if)# ip helper address 10.1.10.1	Enables forwarding and specifies the destination address for forwarding UDP broadcast packets, including BOOTP.

	Command or Action	Purpose
Step 5	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 6	ip forward-protocol {udp [port] nd sdns} Example: Device(config)# ip forward-protocol sdns	Specifies which protocols the router forwards when forwarding broadcast packets.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show ip interface [interface-id] Example: Device# show ip interface gigabitethernet 1/0/1	Verifies the configuration on the interface or all interfaces.
Step 9	show running-config Example: Device# show running-config	Verifies your entries.
Step 10	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Establishing an IP Broadcast Address

The most popular IP broadcast address (and the default) is an address consisting of all ones (255.255.255.255). However, the Device can be configured to generate any form of IP broadcast address.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet 1/0/1</code>	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip broadcast-address <i>ip-address</i> Example: Device(config-if)# <code>ip broadcast-address 128.1.255.255</code>	Enters a broadcast address different from the default, for example 128.1.255.255.
Step 5	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 6	show ip interface [<i>interface-id</i>] Example: Device# <code>show ip interface</code>	Verifies the broadcast address on the interface or all interfaces.
Step 7	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Flooding IP Broadcasts

Procedure

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

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	ip forward-protocol spanning-tree Example: Device(config)# <code>ip forward-protocol spanning-tree</code>	Uses the bridging spanning-tree database to flood UDP datagrams.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 5	show running-config Example: Device# <code>show running-config</code>	Verifies your entries.
Step 6	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.
Step 7	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 8	ip forward-protocol turbo-flood Example: Device(config)# <code>ip forward-protocol turbo-flood</code>	Uses the spanning-tree database to speed up flooding of UDP datagrams.
Step 9	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 10	show running-config Example: Device# <code>show running-config</code>	Verifies your entries.

	Command or Action	Purpose
Step 11	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining IP Addressing

When the contents of a particular cache, table, or database have become or are suspected to be invalid, you can remove all its contents by using the **clear** privileged EXEC commands. The Table lists the commands for clearing contents.

Table 4: Commands to Clear Caches, Tables, and Databases

clear arp-cache	Clears the IP ARP cache and the fast-switching cache.
clear host { <i>name</i> *}	Removes one or all entries from the hostname and the address cache.
clear ip route { <i>network</i> [<i>mask</i>] *}	Removes one or more routes from the IP routing table.

You can display specific statistics, such as the contents of IP routing tables, caches, and databases; the reachability of nodes; and the routing path that packets are taking through the network. The Table lists the privileged EXEC commands for displaying IP statistics.

Table 5: Commands to Display Caches, Tables, and Databases

show arp	Displays the entries in the ARP table.
show hosts	Displays the default domain name, style of lookup service, name server, and the cached list of hostnames and addresses.
show ip aliases	Displays IP addresses mapped to TCP ports (aliases).
show ip arp	Displays the IP ARP cache.
show ip interface [<i>interface-id</i>]	Displays the IP status of interfaces.
show ip irdp	Displays IRDP values.
show ip masks <i>address</i>	Displays the masks used for network addresses and the number of subnets for each mask.
show ip redirects	Displays the address of a default gateway.
show ip route [<i>address</i> [<i>mask</i>]] [<i>protocol</i>]	Displays the current state of the routing table.
show ip route summary	Displays the current state of the routing table in summary form.

How to Configure IP Unicast Routing

Enabling IP Unicast Routing

By default, the Device is in Layer 2 switching mode and IP routing is disabled. To use the Layer 3 capabilities of the Device, you must enable IP routing.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip routing Example: Device(config)# ip routing	Enables IP routing.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show running-config Example: Device# show running-config	Verifies your entries.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example of Enabling IP Routing

This example shows how to enable IP routing :

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing

Device(config-router)# end
```

What to Do Next

You can now set up parameters for the selected routing protocols as described in these sections:

- RIP
- OSPF,
- EIGRP
- BGP
- Unicast Reverse Path Forwarding
- Protocol-Independent Features (optional)

Monitoring and Maintaining the IP Network

You can remove all contents of a particular cache, table, or database. You can also display specific statistics.

Table 6: Commands to Clear IP Routes or Display Route Status

Command	Purpose
<code>show ip route summary</code>	Displays the current state of the routing table in summary form

Feature Information for IP Unicast Routing

Table 7: Feature Information for IP Unicast Routing

Feature Name	Release	Feature Information
IP Unicast Routing	Cisco IOS XE Fuji 16.9.2	This feature was introduced



CHAPTER 4

Configuring RIP

- [Information About RIP, on page 65](#)
- [How to Configure RIP, on page 66](#)
- [Configuration Example for Summary Addresses and Split Horizon, on page 72](#)
- [Feature Information for Routing Information Protocol, on page 73](#)

Information About RIP

The Routing Information Protocol (RIP) is an interior gateway protocol (IGP) created for use in small, homogeneous networks. It is a distance-vector routing protocol that uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. The protocol is documented in RFC 1058. You can find detailed information about RIP in *IP Routing Fundamentals*, published by Cisco Press.



Note RIP is supported in the Network Essentials feature set.

Using RIP, the Device sends routing information updates (advertisements) every 30 seconds. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by that router as unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the non-updating router.

RIP uses hop counts to rate the value of different routes. The hop count is the number of routers that can be traversed in a route. A directly connected network has a hop count of zero; a network with a hop count of 16 is unreachable. This small range (0 to 15) makes RIP unsuitable for large networks.

If the router has a default network path, RIP advertises a route that links the router to the pseudonetwork 0.0.0.0. The 0.0.0.0 network does not exist; it is treated by RIP as a network to implement the default routing feature. The Device advertises the default network if a default was learned by RIP or if the router has a gateway of last resort and RIP is configured with a default metric. RIP sends updates to the interfaces in specified networks. If an interface's network is not specified, it is not advertised in any RIP update.

Summary Addresses and Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about

routes from being advertised by a router on any interface from which that information originated. This feature usually optimizes communication among multiple routers, especially when links are broken.

How to Configure RIP

Default RIP Configuration

Table 8: Default RIP Configuration

Feature	Default Setting
Auto summary	Enabled.
Default-information originate	Disabled.
Default metric	Built-in; automatic metric translations.
IP RIP authentication key-chain	No authentication. Authentication mode: clear text.
IP RIP triggered	Disabled
IP split horizon	Varies with media.
Neighbor	None defined.
Network	None specified.
Offset list	Disabled.
Output delay	0 milliseconds.
Timers basic	<ul style="list-style-type: none"> • Update: 30 seconds. • Invalid: 180 seconds. • Hold-down: 180 seconds. • Flush: 240 seconds.
Validate-update-source	Enabled.
Version	Receives RIP Version 1 and 2 packets; sends Version 1 packets.

Configuring Basic RIP Parameters

To configure RIP, you enable RIP routing for a network and optionally configure other parameters. On the Device, RIP configuration commands are ignored until you configure the network number.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip routing Example: Device(config)# ip routing	Enables IP routing. (Required only if IP routing is disabled.)
Step 4	router rip Example: Device(config)# router rip	Enables a RIP routing process, and enter router configuration mode.
Step 5	network <i>network number</i> Example: Device(config-router)# network 12.0.0.0	Associates a network with a RIP routing process. You can specify multiple network commands. RIP routing updates are sent and received through interfaces only on these networks. Note You must configure a network number for the RIP commands to take effect.
Step 6	neighbor <i>ip-address</i> Example: Device(config-router)# neighbor 10.2.5.1	(Optional) Defines a neighboring router with which to exchange routing information. This step allows routing updates from RIP (normally a broadcast protocol) to reach nonbroadcast networks.
Step 7	offset-list [<i>access-list number name</i>] {<i>in out</i>} <i>offset</i> [<i>type number</i>] Example: Device(config-router)# offset-list 103 in 10	(Optional) Applies an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through RIP. You can limit the offset list with an access list or an interface.
Step 8	timers <i>basic update invalid holddown flush</i> Example: Device(config-router)# timers basic 45 360 400 300	(Optional) Adjusts routing protocol timers. Valid ranges for all timers are 0 to 4294967295 seconds. <ul style="list-style-type: none"> • <i>update</i>—The time between sending routing updates. The default is 30 seconds.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • <i>invalid</i>—The timer after which a route is declared invalid. The default is 180 seconds. • <i>holddown</i>—The time before a route is removed from the routing table. The default is 180 seconds. • <i>flush</i>—The amount of time for which routing updates are postponed. The default is 240 seconds.
Step 9	version {1 2} Example: Device(config-router)# version 2	(Optional) Configures the switch to receive and send only RIP Version 1 or RIP Version 2 packets. By default, the switch receives Version 1 and 2 but sends only Version 1. You can also use the interface commands ip rip {send receive} version 1 2 1 2 to control what versions are used for sending and receiving on interfaces.
Step 10	no auto summary Example: Device(config-router)# no auto summary	(Optional) Disables automatic summarization. By default, the switch summarizes subprefixes when crossing classful network boundaries. Disable summarization (RIP Version 2 only) to advertise subnet and host routing information to classful network boundaries.
Step 11	output-delay delay Example: Device(config-router)# output-delay 8	(Optional) Adds interpacket delay for RIP updates sent. By default, packets in a multiple-packet RIP update have no delay added between packets. If you are sending packets to a lower-speed device, you can add an interpacket delay in the range of 8 to 50 milliseconds.
Step 12	end Example: Device(config-router)# end	Returns to privileged EXEC mode.
Step 13	show ip protocols Example: Device# show ip protocols	Verifies your entries.
Step 14	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface. The key chain specifies the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed, not even the default.

The Device supports two modes of authentication on interfaces for which RIP authentication is enabled: plain text and MD5. The default is plain text.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip rip authentication key-chain <i>name-of-chain</i> Example: Device(config-if)# ip rip authentication key-chain trees	Enables RIP authentication.
Step 5	ip rip authentication mode {text md5} Example: Device(config-if)# ip rip authentication mode md5	Configures the interface to use plain text authentication (the default) or MD5 digest authentication.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show running-config Example: Device# show running-config	Verifies your entries.
Step 8	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# <code>copy running-config startup-config</code>	

Configuring Summary Addresses and Split Horizon



Note In general, disabling split horizon is not recommended unless you are certain that your application requires it to properly advertise routes.

If you want to configure an interface running RIP to advertise a summarized local IP address pool on a network access server for dial-up clients, use the **ip summary-address rip** interface configuration command.



Note If split horizon is enabled, neither autosummary nor interface IP summary addresses are advertised.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet 1/0/1</code>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip address <i>ip-address subnet-mask</i> Example: Device(config-if)# <code>ip address 10.1.1.10 255.255.255.0</code>	Configures the IP address and IP subnet.
Step 5	ip summary-address rip ip address <i>ip-network mask</i> Example:	Configures the IP address to be summarized and the IP network mask.

	Command or Action	Purpose
	Device(config-if)# <code>ip summary-address rip ip address 10.1.1.30 255.255.255.0</code>	
Step 6	no ip split horizon Example: Device(config-if)# <code>no ip split horizon</code>	Disables split horizon on the interface.
Step 7	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 8	show ip interface <i>interface-id</i> Example: Device# <code>show ip interface gigabitethernet 1/0/1</code>	Verifies your entries.
Step 9	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature can optimize communication among multiple routers, especially when links are broken.



Note In general, we do not recommend disabling split horizon unless you are certain that your application requires it to properly advertise routes.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip address <i>ip-address subnet-mask</i> Example: Device(config-if)# ip address 10.1.1.10 255.255.255.0	Configures the IP address and IP subnet.
Step 5	no ip split-horizon Example: Device(config-if)# no ip split-horizon	Disables split horizon on the interface.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show ip interface <i>interface-id</i> Example: Device# show ip interface gigabitethernet 1/0/1	Verifies your entries.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuration Example for Summary Addresses and Split Horizon

In this example, the major net is 10.0.0.0. The summary address 10.2.0.0 overrides the autosummary address of 10.0.0.0 so that 10.2.0.0 is advertised out interface Gigabit Ethernet port 2, and 10.0.0.0 is not advertised. In the example, if the interface is still in Layer 2 mode (the default), you must enter a **no switchport** interface configuration command before entering the **ip address** interface configuration command.



Note If split horizon is enabled, neither autosummary nor interface summary addresses (those configured with the **ip summary-address rip** router configuration command) are advertised.

```
Device(config)# router rip
Device(config-router)# interface gigabitethernet1/0/2
Device(config-if)# ip address 10.1.5.1 255.255.255.0
Device(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Device(config-if)# no ip split-horizon
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 2.2.2.2 peer-group mygroup
Device(config-router)# end
```

Feature Information for Routing Information Protocol

Table 9: Feature Information for IP Unicast Routing

Feature Name	Release	Feature Information
Routing Information Protocol	Cisco IOS XE Fuji 16.9.2	This feature was introduced.



CHAPTER 5

Configuring OSPF

- [Information About OSPF, on page 75](#)
- [How to Configure OSPF, on page 78](#)
- [Monitoring OSPF, on page 88](#)
- [Configuration Examples for OSPF, on page 88](#)
- [Feature Information for OSPF, on page 89](#)

Information About OSPF

OSPF is an Interior Gateway Protocol (IGP) designed expressly for IP networks, supporting IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets. The Cisco implementation supports RFC 1253, OSPF management information base (MIB).

The Cisco implementation conforms to the OSPF Version 2 specifications with these key features:

- Definition of stub areas is supported.
- Routes learned through any IP routing protocol can be redistributed into another IP routing protocol. At the intradomain level, this means that OSPF can import routes learned through EIGRP and RIP. OSPF routes can also be exported into RIP.
- Plain text and MD5 authentication among neighboring routers within an area is supported.
- Configurable routing interface parameters include interface output cost, retransmission interval, interface transmit delay, router priority, router dead and hello intervals, and authentication key.
- Virtual links are supported.
- Not-so-stubby-areas (NSSAs) per RFC 1587 are supported.

OSPF typically requires coordination among many internal routers, area border routers (ABRs) connected to multiple areas, and autonomous system boundary routers (ASBRs). The minimum configuration would use all default parameter values, no authentication, and interfaces assigned to areas. If you customize your environment, you must ensure coordinated configuration of all routers.

OSPF Nonstop Forwarding

The Device or switch stack supports two levels of nonstop forwarding (NSF):

- [OSPF NSF Awareness, on page 76](#)
- [OSPF NSF Capability, on page 76](#)

OSPF NSF Awareness

When the neighboring router is NSF-capable, the Layer 3 Device continues to forward packets from the neighboring router during the interval between the primary Route Processor (RP) in a router crashing and the backup RP taking over, or while the primary RP is manually reloaded for a non-disruptive software upgrade.

This feature cannot be disabled.

OSPF NSF Capability

supports the OSPFv2 NSF IETF format in addition to the OSPFv2 NSF Cisco format that is supported in earlier releases. For information about this feature, see : *NSF—OSPF (RFC 3623 OSPF Graceful Restart)*.

The also supports OSPF NSF-capable routing for IPv4 for better convergence and lower traffic loss following a stack's active switch change.



Note OSPF NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers non-NSF aware neighbors on a network segment, it disables NSF capabilities for that segment. Other network segments where all devices are NSF-aware or NSF-capable continue to provide NSF capabilities.

Use the **nsf** OSPF routing configuration command to enable OSPF NSF routing. Use the **show ip ospf** privileged EXEC command to verify that it is enabled.

For more information, see *Cisco Nonstop Forwarding*:

http://www.cisco.com/en/US/docs/ios/ha/configuration/guide/ha-nonstp_fwdg.html

OSPF Area Parameters

You can optionally configure several OSPF area parameters. These parameters include authentication for password-based protection against unauthorized access to an area, stub areas, and not-so-stubby-areas (NSSAs). Stub areas are areas into which information on external routes is not sent. Instead, the area border router (ABR) generates a default external route into the stub area for destinations outside the autonomous system (AS). An NSSA does not flood all LSAs from the core into the area, but can import AS external routes within the area by redistribution.

Route summarization is the consolidation of advertised addresses into a single summary route to be advertised by other areas. If network numbers are contiguous, you can use the **area range** router configuration command to configure the ABR to advertise a summary route that covers all networks in the range.

Other OSPF Parameters

You can optionally configure other OSPF parameters in router configuration mode.

- **Route summarization:** When redistributing routes from other protocols. Each route is advertised individually in an external LSA. To help decrease the size of the OSPF link state database, you can use the **summary-address** router configuration command to advertise a single router for all the redistributed routes included in a specified network address and mask.

- **Virtual links:** In OSPF, all areas must be connected to a backbone area. You can establish a virtual link in case of a backbone-continuity break by configuring two Area Border Routers as endpoints of a virtual link. Configuration information includes the identity of the other virtual endpoint (the other ABR) and the nonbackbone link that the two routers have in common (the transit area). Virtual links cannot be configured through a stub area.
- **Default route:** When you specifically configure redistribution of routes into an OSPF routing domain, the route automatically becomes an autonomous system boundary router (ASBR). You can force the ASBR to generate a default route into the OSPF routing domain.
- **Domain Name Server (DNS) names for use in all OSPF `show` privileged EXEC command displays** makes it easier to identify a router than displaying it by router ID or neighbor ID.
- **Default Metrics:** OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. The metric is calculated as $ref\text{-}bw$ divided by bandwidth, where ref is 10 by default, and bandwidth (bw) is specified by the **bandwidth** interface configuration command. For multiple links with high bandwidth, you can specify a larger number to differentiate the cost on those links.
- **Administrative distance** is a rating of the trustworthiness of a routing information source, an integer between 0 and 255, with a higher value meaning a lower trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored. OSPF uses three different administrative distances: routes within an area (interarea), routes to another area (interarea), and routes from another routing domain learned through redistribution (external). You can change any of the distance values.
- **Passive interfaces:** Because interfaces between two devices on an Ethernet represent only one network segment, to prevent OSPF from sending hello packets for the sending interface, you must configure the sending device to be a passive interface. Both devices can identify each other through the hello packet for the receiving interface.
- **Route calculation timers:** You can configure the delay time between when OSPF receives a topology change and when it starts the shortest path first (SPF) calculation and the hold time between two SPF calculations.
- **Log neighbor changes:** You can configure the router to send a syslog message when an OSPF neighbor state changes, providing a high-level view of changes in the router.

LSA Group Pacing

The OSPF LSA group pacing feature allows the router to group OSPF LSAs and pace the refreshing, check-summing, and aging functions for more efficient router use. This feature is enabled by default with a 4-minute default pacing interval, and you will not usually need to modify this parameter. The optimum group pacing interval is inversely proportional to the number of LSAs the router is refreshing, check-summing, and aging. For example, if you have approximately 10,000 LSAs in the database, decreasing the pacing interval would benefit you. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might benefit you slightly.

Loopback Interfaces

OSPF uses the highest IP address configured on the interfaces as its router ID. If this interface is down or removed, the OSPF process must recalculate a new router ID and resend all its routing information out its interfaces. If a loopback interface is configured with an IP address, OSPF uses this IP address as its router

ID, even if other interfaces have higher IP addresses. Because loopback interfaces never fail, this provides greater stability. OSPF automatically prefers a loopback interface over other interfaces, and it chooses the highest IP address among all loopback interfaces.

How to Configure OSPF

Default OSPF Configuration

Table 10: Default OSPF Configuration

Feature	Default Setting
Interface parameters	Cost: Retransmit interval: 5 seconds. Transmit delay: 1 second. Priority: 1. Hello interval: 10 seconds. Dead interval: 4 times the hello interval. No authentication. No password specified. MD5 authentication disabled.
Area	Authentication type: 0 (no authentication). Default cost: 1. Range: Disabled. Stub: No stub area defined. NSSA: No NSSA area defined.
Auto cost	100 Mb/s.
Default-information originate	Disabled. When enabled, the default metric setting is 10, and the external route type is Type 2.
Default metric	Built-in, automatic metric translation, as appropriate for each routing protocol.
Distance OSPF	dist1 (all routes within an area): 110. dist2 (all routes from one area to another): 110. dist3 (routes from other routing domains): 110.
OSPF database filter	Disabled. All outgoing link-state advertisements (LSAs) are flooded to the interface.
IP OSPF name lookup	Disabled.
Log adjacency changes	Enabled.
Neighbor	None specified.

Feature	Default Setting
Neighbor database filter	Disabled. All outgoing LSAs are flooded to the neighbor.
Network area	Disabled.
Router ID	No OSPF routing process defined.
Summary address	Disabled.
Timers LSA group pacing	240 seconds.
Timers shortest path first (spf)	spf delay: 50 milliseconds; spf-holdtime: 200 milliseconds.
Virtual link	No area ID or router ID defined. Hello interval: 10 seconds. Retransmit interval: 5 seconds. Transmit delay: 1 second. Dead interval: 40 seconds. Authentication key: no key predefined. Message-digest key (MD5): no key predefined.

Configuring Basic OSPF Parameters

To enable OSPF, create an OSPF routing process, specify the range of IP addresses to associate with the routing process, and assign area IDs to be associated with that range.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router ospf process-id Example: Device(config)# router ospf 15	Enables OSPF routing, and enter router configuration mode. The process ID is an internally used identification parameter that is locally assigned and can be any positive integer. Each OSPF routing process has a unique value. Note OSPF for Routed Access supports only one OSPFv2 and one OSPFv3 instance with a maximum number of 1000 dynamically learned routes.

	Command or Action	Purpose
Step 3	network address wildcard-mask area area-id Example: Device(config) # network 10.1.1.1 255.240.0.0 area 20	Define an interface on which OSPF runs and the area ID for that interface. You can use the wildcard-mask to use a single command to define one or more multiple interfaces to be associated with a specific OSPF area. The area ID can be a decimal value or an IP address.
Step 4	end Example: Device(config) # end	Returns to privileged EXEC mode.
Step 5	show ip protocols Example: Device# show ip protocols	Verifies your entries.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring OSPF Interfaces

You can use the **ip ospf** interface configuration commands to modify interface-specific OSPF parameters. You are not required to modify any of these parameters, but some interface parameters (hello interval, dead interval, and authentication key) must be consistent across all routers in an attached network. If you modify these parameters, be sure all routers in the network have compatible values.



Note The **ip ospf** interface configuration commands are all optional.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface interface-id Example:	Enters interface configuration mode, and specifies the Layer 3 interface to configure.

	Command or Action	Purpose
	Device(config)# interface gigabitethernet 1/0/1	
Step 3	ip ospf cost Example: Device(config-if)# ip ospf 8	(Optional) Explicitly specifies the cost of sending a packet on the interface.
Step 4	ip ospf retransmit-interval seconds Example: Device(config-if)# ip ospf transmit-interval 10	(Optional) Specifies the number of seconds between link state advertisement transmissions. The range is 1 to 65535 seconds. The default is 5 seconds.
Step 5	ip ospf transmit-delay seconds Example: Device(config-if)# ip ospf transmit-delay 2	(Optional) Sets the estimated number of seconds to wait before sending a link state update packet. The range is 1 to 65535 seconds. The default is 1 second.
Step 6	ip ospf priority number Example: Device(config-if)# ip ospf priority 5	(Optional) Sets priority to help find the OSPF designated router for a network. The range is from 0 to 255. The default is 1.
Step 7	ip ospf hello-interval seconds Example: Device(config-if)# ip ospf hello-interval 12	(Optional) Sets the number of seconds between hello packets sent on an OSPF interface. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 10 seconds.
Step 8	ip ospf dead-interval seconds Example: Device(config-if)# ip ospf dead-interval 8	(Optional) Sets the number of seconds after the last device hello packet was seen before its neighbors declare the OSPF router to be down. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 4 times the hello interval.
Step 9	ip ospf authentication-key key Example: Device(config-if)# ip ospf authentication-key password	(Optional) Assign a password to be used by neighboring OSPF routers. The password can be any string of keyboard-entered characters up to 8 bytes in length. All neighboring routers on the same network must have the same password to exchange OSPF information.
Step 10	ip ospf message digest-key keyid md5 key Example: Device(config-if)# ip ospf message digest-key 16 md5 yourlpass	(Optional) Enables MDS authentication. <ul style="list-style-type: none"> • <i>keyid</i>—An identifier from 1 to 255. • <i>key</i>—An alphanumeric password of up to 16 bytes.
Step 11	ip ospf database-filter all out Example:	(Optional) Block flooding of OSPF LSA packets to the interface. By default, OSPF floods new LSAs over all

	Command or Action	Purpose
	Device(config-if)# ip ospf database-filter all out	interfaces in the same area, except the interface on which the LSA arrives.
Step 12	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 13	show ip ospf interface [<i>interface-name</i>] Example: Device# show ip ospf interface	Displays OSPF-related interface information.
Step 14	show ip ospf neighbor detail Example: Device# show ip ospf neighbor detail	Displays NSF awareness status of neighbor switch. The output matches one of these examples: <ul style="list-style-type: none"> • <i>Options is 0x52</i> <i>LLS Options is 0x1 (LR)</i> When both of these lines appear, the neighbor switch is NSF aware. • <i>Options is 0x42</i>—This means the neighbor switch is not NSF aware.
Step 15	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring OSPF Area Parameters

Before you begin



Note The OSPF **area** router configuration commands are all optional.

Procedure

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

	Command or Action	Purpose
	Device# configure terminal	
Step 2	router ospf process-id Example: Device(config)# router ospf 109	Enables OSPF routing, and enter router configuration mode.
Step 3	area area-id authentication Example: Device(config-router)# area 1 authentication	(Optional) Allow password-based protection against unauthorized access to the identified area. The identifier can be either a decimal value or an IP address.
Step 4	area area-id authentication message-digest Example: Device(config-router)# area 1 authentication message-digest	(Optional) Enables MD5 authentication on the area.
Step 5	area area-id stub [no-summary] Example: Device(config-router)# area 1 stub	(Optional) Define an area as a stub area. The no-summary keyword prevents an ABR from sending summary link advertisements into the stub area.
Step 6	area area-id nssa [no-redistribution] [default-information-originate] [no-summary] Example: Device(config-router)# area 1 nssa default-information-originate	(Optional) Defines an area as a not-so-stubby-area. Every router within the same area must agree that the area is NSSA. Select one of these keywords: <ul style="list-style-type: none"> • no-redistribution—Select when the router is an NSSA ABR and you want the redistribute command to import routes into normal areas, but not into the NSSA. • default-information-originate—Select on an ABR to allow importing type 7 LSAs into the NSSA. • no-redistribution—Select to not send summary LSAs into the NSSA.
Step 7	area area-id range address mask Example: Device(config-router)# area 1 range 255.240.0.0	(Optional) Specifies an address range for which a single route is advertised. Use this command only with area border routers.
Step 8	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	show ip ospf [<i>process-id</i>] Example: Device# show ip ospf	Displays information about the OSPF routing process in general or for a specific process ID to verify configuration.
Step 10	show ip ospf [<i>process-id</i> [<i>area-id</i>]] database Example: Device# show ip ospf database	Displays lists of information related to the OSPF database for a specific router.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Other OSPF Parameters

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router ospf <i>process-id</i> Example: Device(config)# router ospf 10	Enables OSPF routing, and enter router configuration mode.
Step 3	summary-address <i>address mask</i> Example: Device(config)# summary-address 10.1.1.1 255.255.255.0	(Optional) Specifies an address and IP subnet mask for redistributed routes so that only one summary route is advertised.
Step 4	area <i>area-id</i> virtual-link <i>router-id</i> [hello-interval <i>seconds</i>] [retransmit-interval <i>seconds</i>] [trans] [[authentication-key <i>key</i>] message-digest-key <i>keyid</i> md5 <i>key</i>]] Example: Device(config)# area 2 virtual-link 192.168.255.1 hello-interval 5	(Optional) Establishes a virtual link and set its parameters.

	Command or Action	Purpose
Step 5	<p>default-information originate [always] [metric <i>metric-value</i>] [metric-type <i>type-value</i>] [route-map <i>map-name</i>]</p> <p>Example:</p> <pre>Device(config)# default-information originate metric 100 metric-type 1</pre>	(Optional) Forces the ASBR to generate a default route into the OSPF routing domain. Parameters are all optional.
Step 6	<p>ip ospf name-lookup</p> <p>Example:</p> <pre>Device(config)# ip ospf name-lookup</pre>	(Optional) Configures DNS name lookup. The default is disabled.
Step 7	<p>ip auto-cost reference-bandwidth <i>ref-bw</i></p> <p>Example:</p> <pre>Device(config)# ip auto-cost reference-bandwidth 5</pre>	(Optional) Specifies an address range for which a single route will be advertised. Use this command only with area border routers.
Step 8	<p>distance ospf {[inter-area <i>dist1</i>] [inter-area <i>dist2</i>] [external <i>dist3</i>]}</p> <p>Example:</p> <pre>Device(config)# distance ospf inter-area 150</pre>	(Optional) Changes the OSPF distance values. The default distance for each type of route is 110. The range is 1 to 255.
Step 9	<p>passive-interface <i>type number</i></p> <p>Example:</p> <pre>Device(config)# passive-interface gigabitethernet 1/0/6</pre>	(Optional) Suppresses the sending of hello packets through the specified interface.
Step 10	<p>timers throttle spf <i>spf-delay</i> <i>spf-holdtime</i> <i>spf-wait</i></p> <p>Example:</p> <pre>Device(config)# timers throttle spf 200 100 100</pre>	<p>(Optional) Configures route calculation timers.</p> <ul style="list-style-type: none"> • <i>spf-delay</i>—Delay between receiving a change to SPF calculation. The range is from 1 to 600000 in milliseconds. • <i>spf-holdtime</i>—Delay between first and second SPF calculation. The range is from 1 to 600000 in milliseconds. • <i>spf-wait</i>—Maximum wait time in milliseconds for SPF calculations. The range is from 1 to 600000 in milliseconds.
Step 11	<p>ospf log-adj-changes</p> <p>Example:</p> <pre>Device(config)# ospf log-adj-changes</pre>	(Optional) Sends syslog message when a neighbor state changes.

	Command or Action	Purpose
Step 12	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 13	show ip ospf [process-id [area-id]] database Example: Device# show ip ospf database	Displays lists of information related to the OSPF database for a specific router.
Step 14	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Changing LSA Group Pacing

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router ospf process-id Example: Device(config)# router ospf 25	Enables OSPF routing, and enter router configuration mode.
Step 3	timers lsa-group-pacing seconds Example: Device(config-router)# timers lsa-group-pacing 15	Changes the group pacing of LSAs.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show running-config Example:	Verifies your entries.

	Command or Action	Purpose
	Device# <code>show running-config</code>	
Step 6	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring a Loopback Interface

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	interface loopback 0 Example: Device(config)# <code>interface loopback 0</code>	Creates a loopback interface, and enter interface configuration mode.
Step 3	ip address address mask Example: Device(config-if)# <code>ip address 10.1.1.5 255.255.240.0</code>	Assign an IP address to this interface.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 5	show ip interface Example: Device# <code>show ip interface</code>	Verifies your entries.
Step 6	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

Command or Action	Purpose
Device# <code>copy running-config startup-config</code>	

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases.

Table 11: Show IP OSPF Statistics Commands

<code>show ip ospf [process-id]</code>	Displays general information about OSPF processes.
<code>show ip ospf [process-id] database [router] [link-state-id]</code> <code>show ip ospf [process-id] database [router] [self-originate]</code> <code>show ip ospf [process-id] database [router] [adv-router [ip-address]]</code> <code>show ip ospf [process-id] database [network] [link-state-id]</code> <code>show ip ospf [process-id] database [summary] [link-state-id]</code> <code>show ip ospf [process-id] database [asbr-summary] [link-state-id]</code> <code>show ip ospf [process-id] database [external] [link-state-id]</code> <code>show ip ospf [process-id area-id] database [database-summary]</code>	Displays lists of information about OSPF databases.
<code>show ip ospf border-routes</code>	Displays the internal OSPF routing table entries.
<code>show ip ospf interface [interface-name]</code>	Displays OSPF-related information for the specified interface.
<code>show ip ospf neighbor [interface-name] [neighbor-id] detail</code>	Displays OSPF neighbor information for the specified interface.
<code>show ip ospf virtual-links</code>	Displays OSPF-related information for virtual links.

Configuration Examples for OSPF

Example: Configuring Basic OSPF Parameters

This example shows how to configure an OSPF routing process and assign it a process number of 109:

```
Device(config)# router ospf 109
Device(config-router)# network 131.108.0.0 255.255.255.0 area 24
```

Feature Information for OSPF

Table 12: Feature Information for OSPF

Feature Name	Release	Feature Information
OSPF (Open Shortest Path First)	Cisco IOS XE Fuji 16.9.2	This feature was introduced



CHAPTER 6

Configuring EIGRP

- [Information About EIGRP, on page 91](#)
- [How to Configure EIGRP, on page 94](#)
- [Monitoring and Maintaining EIGRP, on page 101](#)
- [Feature Information for EIGRP, on page 101](#)

Information About EIGRP

Enhanced IGRP (EIGRP) is a Cisco proprietary enhanced version of the IGRP. EIGRP uses the same distance vector algorithm and distance information as IGRP; however, the convergence properties and the operating efficiency of EIGRP are significantly improved.

The convergence technology employs an algorithm referred to as the Diffusing Update Algorithm (DUAL), which guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations.

IP EIGRP provides increased network width. With RIP, the largest possible width of your network is 15 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport-layer hop counter. EIGRP increments the transport control field only when an IP packet has traversed 15 routers and the next hop to the destination was learned through EIGRP. When a RIP route is used as the next hop to the destination, the transport control field is incremented as usual.

EIGRP Features

EIGRP offers these features:

- Fast convergence.
- Incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table, minimizing the bandwidth required for EIGRP packets.
- Less CPU usage because full update packets need not be processed each time they are received.
- Protocol-independent neighbor discovery mechanism to learn about neighboring routers.
- Variable-length subnet masks (VLSMs).
- Arbitrary route summarization.

- EIGRP scales to large networks.

EIGRP Components

EIGRP has these four basic components:

- Neighbor discovery and recovery is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery and recovery is achieved with low overhead by periodically sending small hello packets. As long as hello packets are received, the Cisco IOS software can learn that a neighbor is alive and functioning. When this status is determined, the neighboring routers can exchange routing information.
- The reliable transport protocol is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably, and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet), it is not necessary to send hellos reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is shown in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.
- The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as a metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors, but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL tests for feasible successors. If there are feasible successors, it uses any it finds to avoid unnecessary recomputation.
- The protocol-dependent modules are responsible for network layer protocol-specific tasks. An example is the IP EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in IP. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IP routing table. EIGRP is also responsible for redistributing routes learned by other IP routing protocols.



Note To enable EIGRP, the Device or active switch must be running the

EIGRP Nonstop Forwarding

The Device stack supports two levels of EIGRP nonstop forwarding:

- EIGRP NSF Awareness

- EIGRP NSF Capability

EIGRP NSF Awareness

The supports EIGRP NSF Awareness for IPv4. When the neighboring router is NSF-capable, the Layer 3 Device continues to forward packets from the neighboring router during the interval between the primary Route Processor (RP) in a router failing and the backup RP taking over, or while the primary RP is manually reloaded for a nondisruptive software upgrade. This feature cannot be disabled.

EIGRP NSF Capability

The supports EIGRP Cisco NSF routing to speed up convergence and to eliminate traffic loss after a stack's active switch changeover.

The also supports EIGRP NSF-capable routing for IPv4 for better convergence and lower traffic loss following an active switch changeover. When an EIGRP NSF-capable active switch restarts or a new active switch starts up and NSF restarts, the Device has no neighbors, and the topology table is empty. The Device must bring up the interfaces, reacquire neighbors, and rebuild the topology and routing tables without interrupting the traffic directed toward the Device stack. EIGRP peer routers maintain the routes learned from the new active switch and continue forwarding traffic through the NSF restart process.

To prevent an adjacency reset by the neighbors, the new active switch uses a new Restart (RS) bit in the EIGRP packet header to show the restart. When the neighbor receives this, it synchronizes the stack in its peer list and maintains the adjacency with the stack. The neighbor then sends its topology table to the active switch with the RS bit set to show that it is NSF-aware and is aiding the new active switch.

If at least one of the stack peer neighbors is NSF-aware, the active switch receives updates and rebuilds its database. Each NSF-aware neighbor sends an end of table (EOT) marker in the last update packet to mark the end of the table content. The active switch recognizes the convergence when it receives the EOT marker, and it then begins sending updates. When the active switch has received all EOT markers from its neighbors or when the NSF converge timer expires, EIGRP notifies the routing information database (RIB) of convergence and floods its topology table to all NSF-aware peers.

EIGRP Stub Routing

The EIGRP stub routing feature reduces resource utilization by moving routed traffic closer to the end user.



Note The device uses EIGRP stub routing at the access layer to eliminate the need for other types of routing advertisements.

EIGRP stub routing is supported with the Network Essentials license.

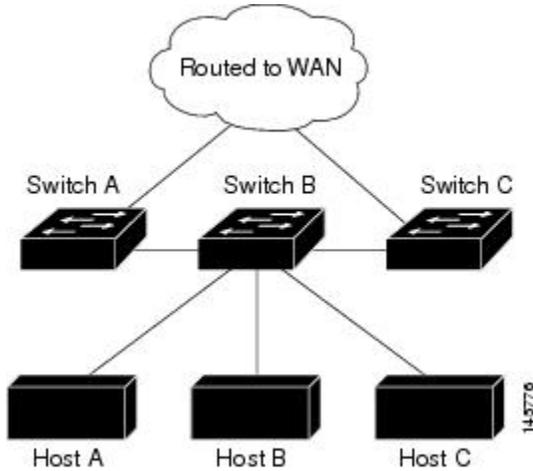
In a network using EIGRP stub routing, the only allowable route for IP traffic to the user is through a device that is configured with EIGRP stub routing. The device sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRP stub routing, you need to configure the distribution and remote routers to use EIGRP and to configure only the device as a stub. Only specified routes are propagated from the device. The device responds to all queries for summaries, connected routes, and routing updates.

Any neighbor that receives a packet informing it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

In the figure given below, device B is configured as an EIGRP stub router. Devices A and C are connected to the rest of the WAN. Device B advertises connected, static, redistribution, and summary routes to Device A and C. Device B does not advertise any routes learned from Device A (and the reverse).

Figure 4: EIGRP Stub Router Configuration



How to Configure EIGRP

To create an EIGRP routing process, you must enable EIGRP and associate networks. EIGRP sends updates to the interfaces in the specified networks. If you do not specify an interface network, it is not advertised in any EIGRP update.



Note If you have routers on your network that are configured for IGRP, and you want to change to EIGRP, you must designate transition routers that have both IGRP and EIGRP configured. In these cases, perform Steps 1 through 3 in the next section and also see the “Configuring Split Horizon” section. You must use the same AS number for routes to be automatically redistributed.

Default EIGRP Configuration

Table 13: Default EIGRP Configuration

Feature	Default Setting
Auto summary	Disabled.
Default-information	Exterior routes are accepted and default information is passed between processes when doing redistribution.

Feature	Default Setting
Default metric	Only connected routes and interface static routes can be redistributed with the default metric. The metric includes: <ul style="list-style-type: none"> • Bandwidth: 0 or greater kb/s. • Delay (tens of microseconds): 0 or any positive number that is less than or equal to 39.1 nanoseconds. • Reliability: any number between 0 and 255 (255 means 100 percent reliability). • Loading: effective bandwidth as a number between 0 and 255 (255 means 100 percent loading). • MTU: maximum transmission unit size of the route in bytes. 0 or greater integer.
Distance	Internal distance: 90. External distance: 170.
EIGRP log-neighbor changes	Disabled. No adjacency changes logged.
IP authentication key-chain	No authentication provided.
IP authentication mode	No authentication provided.
IP bandwidth-percent	50 percent.
IP hello interval	For low-speed nonbroadcast multiaccess (NBMA) networks: 60 seconds. For all other networks: 5 seconds.
IP hold-time	For low-speed NBMA networks: 180 seconds; all other networks: 15 seconds.
IP split-horizon	Enabled.
IP summary address	No summary aggregate addresses are predefined.
Metric weights	tos: 0; k1 and k3: 1; k2, k4, and k5: 0
Network	None specified.
Nonstop Forwarding (NSF) Awareness	Enabled for IPv4 on switches running the Allows Layer 3 switches to continue forwarding packets from a neighboring NSF-capable router during software changes.
NSF capability	Disabled. Note The Device supports EIGRP NSF-capable routing for IPv4.
Offset-list	Disabled.
Router EIGRP	Disabled.
Set metric	No metric set in the route map.

Feature	Default Setting
Traffic-share	Distributed proportionately to the ratios of the metrics.
Variance	1 (equal-cost load-balancing).

Configuring Basic EIGRP Parameters

Procedure

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>router eigrp autonomous-system</p> <p>Example:</p> <pre>Device(config)# router eigrp 10</pre>	Enables an EIGRP routing process, and enter router configuration mode. The AS number identifies the routes to other EIGRP routers and is used to tag routing information.
Step 3	<p>nsf</p> <p>Example:</p> <pre>Device(config-router)# nsf</pre>	(Optional) Enables EIGRP NSF. Enter this command on the active switch and on all of its peers.
Step 4	<p>network network-number</p> <p>Example:</p> <pre>Device(config-router)# network 192.168.0.0</pre>	Associate networks with an EIGRP routing process. EIGRP sends updates to the interfaces in the specified networks.
Step 5	<p>eigrp log-neighbor-changes</p> <p>Example:</p> <pre>Device(config-router)# eigrp log-neighbor-changes</pre>	(Optional) Enables logging of EIGRP neighbor changes to monitor routing system stability.
Step 6	<p>metric weights <i>tos k1 k2 k3 k4 k5</i></p> <p>Example:</p> <pre>Device(config-router)# metric weights 0 2 0 2 0 0</pre>	<p>(Optional) Adjust the EIGRP metric. Although the defaults have been carefully set to provide excellent operation in most networks, you can adjust them.</p> <p>Caution Setting metrics is complex and is not recommended without guidance from an experienced network designer.</p>
Step 7	<p>offset-list [<i>access-list number name</i>] {in out} <i>offset</i> [<i>type number</i>]</p>	(Optional) Applies an offset list to routing metrics to increase incoming and outgoing metrics to routes learned

	Command or Action	Purpose
	Example: Device(config-router)# offset-list 21 out 10	through EIGRP. You can limit the offset list with an access list or an interface.
Step 8	auto-summary Example: Device(config-router)# auto-summary	(Optional) Enables automatic summarization of subnet routes into network-level routes.
Step 9	interface interface-id Example: Device(config-router)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 10	ip summary-address eigrp autonomous-system-number address mask Example: Device(config-if)# ip summary-address eigrp 1 192.168.0.0 255.255.0.0	(Optional) Configures a summary aggregate.
Step 11	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 12	show ip protocols Example: Device# show ip protocols	Verifies your entries. For NSF awareness, the output shows: *** IP Routing is NSF aware *** EIGRP NSF enabled
Step 13	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring EIGRP Interfaces

Other optional EIGRP parameters can be configured on an interface basis.

Procedure

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>interface interface-id</p> <p>Example:</p> <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 3	<p>ip bandwidth-percent eigrp percent</p> <p>Example:</p> <pre>Device(config-if)# ip bandwidth-percent eigrp 60</pre>	(Optional) Configures the percentage of bandwidth that can be used by EIGRP on an interface. The default is 50 percent.
Step 4	<p>ip summary-address eigrp autonomous-system-number address mask</p> <p>Example:</p> <pre>Device(config-if)# ip summary-address eigrp 109 192.161.0.0 255.255.0.0</pre>	(Optional) Configures a summary aggregate address for a specified interface (not usually necessary if auto-summary is enabled).
Step 5	<p>ip hello-interval eigrp autonomous-system-number seconds</p> <p>Example:</p> <pre>Device(config-if)# ip hello-interval eigrp 109 10</pre>	(Optional) Change the hello time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 60 seconds for low-speed NBMA networks and 5 seconds for all other networks.
Step 6	<p>ip hold-time eigrp autonomous-system-number seconds</p> <p>Example:</p> <pre>Device(config-if)# ip hold-time eigrp 109 40</pre>	<p>(Optional) Change the hold time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 180 seconds for low-speed NBMA networks and 15 seconds for all other networks.</p> <p>Caution Do not adjust the hold time without consulting Cisco technical support.</p>
Step 7	<p>no ip split-horizon eigrp autonomous-system-number</p> <p>Example:</p> <pre>Device(config-if)# no ip split-horizon eigrp 109</pre>	(Optional) Disables split horizon to allow route information to be advertised by a router out any interface from which that information originated.
Step 8	<p>end</p> <p>Example:</p>	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# end	
Step 9	show ip eigrp interface Example: Device# show ip eigrp interface	Displays which interfaces EIGRP is active on and information about EIGRP relating to those interfaces.
Step 10	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring EIGRP Route Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol to prevent the introduction of unauthorized or false routing messages from unapproved sources.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 3	ip authentication mode eigrp autonomous-system md5 Example: Device(config-if)# ip authentication mode eigrp 104 md5	Enables MD5 authentication in IP EIGRP packets.
Step 4	ip authentication key-chain eigrp autonomous-system key-chain Example: Device(config-if)# ip authentication key-chain eigrp 105 chain1	Enables authentication of IP EIGRP packets.

	Command or Action	Purpose
Step 5	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Returns to global configuration mode.
Step 6	<p>key chain <i>name-of-chain</i></p> <p>Example:</p> <pre>Device(config)# key chain chain1</pre>	Identify a key chain and enter key-chain configuration mode. Match the name configured in Step 4.
Step 7	<p>key number</p> <p>Example:</p> <pre>Device(config-keychain)# key 1</pre>	In key-chain configuration mode, identify the key number.
Step 8	<p>key-string <i>text</i></p> <p>Example:</p> <pre>Device(config-keychain-key)# key-string key1</pre>	In key-chain key configuration mode, identify the key string.
Step 9	<p>accept-lifetime <i>start-time</i> {infinite <i>end-time</i> duration <i>seconds</i>}</p> <p>Example:</p> <pre>Device(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2011 duration 7200</pre>	<p>(Optional) Specifies the time period during which the key can be received.</p> <p>The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i>. The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite.</p>
Step 10	<p>send-lifetime <i>start-time</i> {infinite <i>end-time</i> duration <i>seconds</i>}</p> <p>Example:</p> <pre>Device(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2011 duration 3600</pre>	<p>(Optional) Specifies the time period during which the key can be sent.</p> <p>The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i>. The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite.</p>
Step 11	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 12	<p>show key chain</p> <p>Example:</p> <pre>Device# show key chain</pre>	Displays authentication key information.

	Command or Action	Purpose
Step 13	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining EIGRP

You can delete neighbors from the neighbor table. You can also display various EIGRP routing statistics. The table given below lists the privileged EXEC commands for deleting neighbors and displaying statistics.

Table 14: IP EIGRP Clear and Show Commands

<code>clear ip eigrp neighbors [if-address interface]</code>	Deletes neighbors from the neighbor table.
<code>show ip eigrp interface [interface] [as number]</code>	Displays information about interfaces participating in EIGRP.
<code>show ip eigrp neighbors [type-number]</code>	Displays EIGRP discovered neighbors.
<code>show ip eigrp topology [autonomous-system-number] [[ip-address] mask]</code>	Displays the EIGRP topology table.
<code>show ip eigrp traffic [autonomous-system-number]</code>	Displays the number of packets sent and received by the EIGRP process.

Feature Information for EIGRP

Table 15: Feature Information for EIGRP

Feature Name	Release	Feature Information
EIGRP (Enhanced IGRP)	Cisco IOS XE Fuji 16.9.2	This feature was introduced.



CHAPTER 7

Configuring IS-IS Routing

- [Information About IS-IS Routing, on page 103](#)
- [How to Configure IS-IS, on page 105](#)
- [Monitoring and Maintaining IS-IS, on page 113](#)
- [Feature Information for IS-IS, on page 113](#)

Information About IS-IS Routing

Integrated Intermediate System-to-Intermediate System (IS-IS) is an ISO dynamic routing protocol (described in ISO 105890). To enable IS-IS you should create an IS-IS routing process and assign it to a specific interface, rather than to a network. You can specify more than one IS-IS routing process per Layer 3 device by using the multiarea IS-IS configuration syntax. You should then configure the parameters for each instance of the IS-IS routing process.

Small IS-IS networks are built as a single area that includes all the devices in the network. As the network grows larger, the network reorganizes itself into a backbone area made up of all the connected set of Level 2 devices still connected to their local areas. Within a local area, devices know how to reach all system IDs. Between areas, devices know how to reach the backbone, and the backbone devices know how to reach other areas.

Devices establish Level 1 adjacencies to perform routing within a local area (station routing). Devices establish Level 2 adjacencies to perform routing between Level 1 areas (area routing).

A single Cisco device can participate in routing in up to 29 areas and can perform Level 2 routing in the backbone. In general, each routing process corresponds to an area. By default, the first instance of the routing process that is configured performs both Level 1 and Level 2 routing. You can configure additional device instances, which are automatically treated as Level 1 areas. You must configure the parameters for each instance of the IS-IS routing process individually.

For IS-IS multiarea routing, you can configure only one process to perform Level 2 routing, although you can define up to 29 Level 1 areas for each Cisco unit. If Level 2 routing is configured on any process, all additional processes are automatically configured as Level 1. You can configure this process to perform Level 1 routing at the same time. If Level 2 routing is not desired for a device instance, remove the Level 2 capability using the **is-type** command in global configuration mode. Use the **is-type** command also to configure a different device instance as a Level 2 device.

Nonstop Forwarding Awareness

The integrated IS-IS Nonstop Forwarding (NSF) Awareness feature is supported for IPv4G. The feature allows customer premises equipment (CPE) devices that are NSF-aware to help NSF-capable devices perform nonstop forwarding of packets. The local device is not necessarily performing NSF, but its NSF awareness capability allows the integrity and accuracy of the routing database and the link-state database on the neighboring NSF-capable device to be maintained during the switchover process.

The integrated IS-IS Nonstop Forwarding (NSF) Awareness feature is automatically enabled and requires no configuration.

IS-IS Global Parameters

The following are the optional IS-IS global parameters that you can configure:

- You can force a default route into an IS-IS routing domain by configuring a default route that is controlled by a route map. You can also specify the other filtering options that are configurable under a route map.
- You can configure the device to ignore IS-IS link-state packets (LSPs) that are received with internal checksum errors, or to purge corrupted LSPs, and cause the initiator of the LSP to regenerate it.
- You can assign passwords to areas and domains.
- You can create aggregate addresses that are represented in the routing table by a summary address (based on route summarization). Routes learned from other routing protocols can also be summarized. The metric used to advertise the summary is the smallest metric of all the specific routes.
- You can set an overload bit.
- You can configure the LSP refresh interval and the maximum time that an LSP can remain in the device database without a refresh.
- You can set the throttling timers for LSP generation, shortest path first computation, and partial route computation.
- You can configure the device to generate a log message when an IS-IS adjacency changes state (Up or Down).
- If a link in the network has a maximum transmission unit (MTU) size of less than 1500 bytes, you can lower the LSP MTU so that routing still occurs.
- You can use the **partition avoidance** command to prevent an area from becoming partitioned when full connectivity is lost among a Level 1-2 border device, adjacent Level 1 devices, and end hosts.

IS-IS Interface Parameters

You can optionally configure certain interface-specific IS-IS parameters independently from other attached devices. However, if you change default value, such as multipliers and time intervals, it makes sense to also change them on multiple devices and interfaces. Most of the interface parameters can be configured for level 1, level 2, or both.

The following are the interface-level parameters that you can configure:

- The default metric on the interface that is used as a value for the IS-IS metric and assigned when quality of service (QoS) routing is not performed.

- The hello interval (length of time between hello packets sent on the interface) or the default hello packet multiplier used on the interface to determine the hold time sent in IS-IS hello packets. The hold time determines how long a neighbor waits for another hello packet before declaring the neighbor down. This determines how quickly a failed link or neighbor is detected so that routes can be recalculated. Change the hello multiplier in circumstances where hello packets are lost frequently and IS-IS adjacencies are failing unnecessarily. You can raise the hello multiplier and lower the hello interval correspondingly to make the hello protocol more reliable, without increasing the time required to detect a link failure.
- Other time intervals:
 - Complete sequence number PDU (CSNP) interval—CSNPs are sent by the designated device to maintain database synchronization.
 - Retransmission interval—This is the time between retransmission of IS-IS LSPs for point-to-point links.
 - IS-IS LSP retransmission throttle interval—This is the maximum rate (number of milliseconds between packets) at which IS-IS LSPs are resent on point-to-point links. This interval is different from the retransmission interval, which is the time between successive retransmissions of the same LSP.
- Designated device-election priority, which allows you to reduce the number of adjacencies required on a multiaccess network, which in turn reduces the amount of routing protocol traffic and the size of the topology database.
- The interface circuit type, which is the type of adjacency required for neighbors on the specified interface.
- Password authentication for the interface.

How to Configure IS-IS

The following sections provide information on how to enable IS-IS on an interface, how to configure IS-IS global parameters, and how to configure IS-IS interface parameters.

Default IS-IS Configuration

Table 16: Default IS-IS Configuration

Feature	Default Setting
Ignore link-state PDU (LSP) errors	Enabled.
IS-IS type	Conventional IS-IS—The router acts as both a Level 1 (station) and a router. Multiarea IS-IS—The first instance of the IS-IS routing process is a router. Remaining instances are Level 1 routers.
Default-information originate	Disabled.
Log IS-IS adjacency state changes.	Disabled.

Feature	Default Setting
LSP generation throttling timers	Maximum interval between two consecutive occurrences—5000 milliseconds. Initial LSP generation delay—50 milliseconds. Hold time between the first and second LSP generation—200 milliseconds.
LSP maximum lifetime (without a refresh)	1200 seconds (20 minutes) before the LSP packet is deleted.
LSP refresh interval	Every 900 seconds (15 minutes).
Maximum LSP packet size	1497 bytes.
NSF Awareness	Enabled. Allows Layer 3 devices to continue forwarding packets from a nonstop forwarding-capable router during hardware or software changes.
Partial route computation (PRC) throttling timers	Maximum PRC wait interval—5000 milliseconds. Initial PRC calculation delay after a topology change—50 milliseconds. Hold time between the first and second PRC calculation—200 milliseconds.
Partition avoidance	Disabled.
Password	No area or domain password is defined, and authentication is disabled.
Set-overload-bit	Disabled. When enabled, if no arguments are entered, the overload bit is set immediately and remains set until you enter the no set-overload-bit command.
Shortest path first (SPF) throttling timers	Maximum interval between consecutive SFPS—5000 milliseconds. Initial SPF calculation after a topology change—200 milliseconds. Hold time between the first and second SPF calculation—50 milliseconds.
Summary-address	Disabled.

Enabling IS-IS Routing

To enable IS-IS, specify a name and a network entity title (NET) for each routing process. Enable IS-IS routing on the interface and specify the area for each instance of the routing process.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router isis [area tag] Example:	Enables IS-IS routing for the specified routing process and enters IS-IS routing configuration mode.

	Command or Action	Purpose
	<pre>Device(config)# router isis tag1</pre>	<p>(Optional) Use the <i>area tag</i> argument to identify the area to which the IS-IS router is assigned. Enter a value if you are configuring multiple IS-IS areas.</p> <p>The first IS-IS instance that is configured is Level 1-2 by default. Later instances are automatically configured as Level 1. You can change the level of routing by using the is-type command in global configuration mode.</p>
Step 3	<p>net <i>network-entity-title</i></p> <p>Example:</p> <pre>Device(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00</pre>	Configures the NETs for the routing process. While configuring multiarea IS-IS, specify a NET for each routing process. Specify a name for a NET and for an address.
Step 4	<p>is-type {level-1 level-1-2 level-2-only}</p> <p>Example:</p> <pre>Device(config-router)# is-type level-2-only</pre>	<p>(Optional) Configures the router to act as a Level 1 (station) router, a Level 2 (area) router for multiarea routing, or both (the default):</p> <ul style="list-style-type: none"> • level 1—Acts as a station router only. • level 1-2—Acts as both a station router and an area router. • level 2—Acts as an area router only.
Step 5	<p>exit</p> <p>Example:</p> <pre>Device(config-router)# end</pre>	Returns to global configuration mode.
Step 6	<p>interface <i>interface-id</i></p> <p>Example:</p> <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Specifies an interface to route IS-IS, and enters interface configuration mode. If the interface is not already configured as a Layer 3 interface, enter the no switchport command to configure the interface into Layer 3 mode.
Step 7	<p>ip router isis [<i>area tag</i>]</p> <p>Example:</p> <pre>Device(config-if)# ip router isis tag1</pre>	Configures an IS-IS routing process on the interface and attaches an area designator to the routing process.
Step 8	<p>ip address <i>ip-address-mask</i></p> <p>Example:</p> <pre>Device(config-if)# ip address 10.0.0.5 255.255.255.0</pre>	Defines the IP address for the interface. An IP address is required for all the interfaces in an area, that is enabled for IS-IS, if any one interface is configured for IS-IS routing.
Step 9	<p>end</p> <p>Example:</p>	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# end	
Step 10	show isis [area tag] database detail Example: Device# show isis database detail	Verifies your entries.

Configuring IS-IS Global Parameters

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router isis Example: Device(config)# router isis	Specifies the IS-IS routing protocol and enters router configuration mode.
Step 3	default-information originate [route-map map-name] Example: Device(config-router)# default-information originate route-map map1	(Optional) Forces a default route into the IS-IS routing domain. If you enter route-map map-name , the routing process generates the default route if the route map is satisfied.
Step 4	ignore-lsp-errors Example: Device(config-router)# ignore-lsp-errors	(Optional) Configures the router to ignore LSPs with internal checksum errors, instead of purging the LSPs. This command is enabled by default (corrupted LSPs are dropped). To purge the corrupted LSPs, enter the no ignore-lsp-errors router configuration command.
Step 5	area-password password Example: Device(config-router)# area-password 1password	(Optional) Configures the area authentication password that is inserted in Level 1 (station router level) LSPs.
Step 6	domain-password password Example: Device(config-router)# domain-password 2password	(Optional) Configures the routing domain authentication password that is inserted in Level 2 (area router level) LSPs.

	Command or Action	Purpose
Step 7	<p>summary-address <i>address mask</i> [level-1 level-1-2 level-2]</p> <p>Example:</p> <pre>Device(config-router)# summary-address 10.1.0.0 255.255.0.0 level-2</pre>	(Optional) Creates a summary of addresses for a given level.
Step 8	<p>set-overload-bit [on-startup {<i>seconds</i> wait-for-bgp}]</p> <p>Example:</p> <pre>Device(config-router)# set-overload-bit on-startup wait-for-bgp</pre>	<p>(Optional) Sets an overload bit to allow other routers to ignore the router in their shortest path first (SPF) calculations if the router is having problems.</p> <ul style="list-style-type: none"> • (Optional) on-startup—Sets the overload bit only on startup. If on-startup is not specified, the overload bit is set immediately and remains set until you enter the no set-overload-bit command. If on-startup is specified, you must either enter number of seconds or enter wait-for-bgp. • <i>seconds</i>—When the on-startup keyword is configured, it causes the overload bit to be set when the system is started and remains set for the specified number of seconds. The range is from 5 to 86400 seconds. • wait-for-bgp—When the on-startup keyword is configured, it causes the overload bit to be set when the system is started and remains set until BGP has converged. If BGP does not signal the IS-IS that it is converged, the IS-IS will turn off the overload bit after 10 minutes.
Step 9	<p>lsp-refresh-interval <i>seconds</i></p> <p>Example:</p> <pre>Device(config-router)# lsp-refresh-interval 1080</pre>	(Optional) Sets an LSP refresh interval, in seconds. The range is from 1 to 65535 seconds. The default is to send LSP refreshes every 900 seconds (15 minutes).
Step 10	<p>max-lsp-lifetime <i>seconds</i></p> <p>Example:</p> <pre>Device(config-router)# max-lsp-lifetime 1000</pre>	(Optional) Sets the maximum time that LSP packets remain in the router database without being refreshed. The range is from 1 to 65535 seconds. The default is 1200 seconds (20 minutes). After the specified time interval, the LSP packet is deleted.
Step 11	<p>lsp-gen-interval [level-1 level-2] <i>lsp-max-wait</i> [<i>lsp-initial-wait</i> <i>lsp-second-wait</i>]</p> <p>Example:</p> <pre>Device(config-router)# lsp-gen-interval level-2 2 50 100</pre>	<p>(Optional) Sets the IS-IS LSP generation throttling timers:</p> <ul style="list-style-type: none"> • <i>lsp-max-wait</i>—Maximum interval (in milliseconds) between two consecutive occurrences of an LSP being generated. The range is from 1 to 120; the default is 5000.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • <i>lsp-initial-wait</i>—Initial LSP generation delay (in milliseconds). The range is from 1 to 10000; the default is 50. • <i>lsp-second-wait</i>—Hold time between the first and second LSP generation (in milliseconds). The range is from 1 to 10000; the default is 200.
Step 12	<p>spf-interval [level-1 level-2] <i>spf-max-wait</i> [<i>spf-initial-wait</i> <i>spf-second-wait</i>]</p> <p>Example:</p> <pre>Device(config-router)# spf-interval level-2 5 10 20</pre>	<p>(Optional) Sets IS-IS SPF throttling timers.</p> <ul style="list-style-type: none"> • <i>spf-max-wait</i>—Maximum interval between consecutive SFPs (in milliseconds). The range is from 1 to 120; the default is 5000. • <i>spf-initial-wait</i>—Initial SFP calculation after a topology change (in milliseconds). The range is from 1 to 10000; the default is 50. • <i>spf-second-wait</i>—Hold time between the first and second SFP calculation (in milliseconds). The range is from 1 to 10000; the default is 200.
Step 13	<p>prc-interval <i>prc-max-wait</i> [<i>prc-initial-wait</i> <i>prc-second-wait</i>]</p> <p>Example:</p> <pre>Device(config-router)# prc-interval 5 10 20</pre>	<p>(Optional) Sets IS-IS PRC throttling timers.</p> <ul style="list-style-type: none"> • <i>prc-max-wait</i>—Maximum interval (in milliseconds) between two consecutive PRC calculations. The range is from 1 to 120; the default is 5000. • <i>prc-initial-wait</i>—Initial PRC calculation delay (in milliseconds) after a topology change. The range is from 1 to 10,000; the default is 50. • <i>prc-second-wait</i>—Hold time between the first and second PRC calculation (in milliseconds). The range is 1 to 10,000; the default is 200.
Step 14	<p>log-adjacency-changes [all]</p> <p>Example:</p> <pre>Device(config-router)# log-adjacency-changes all</pre>	<p>(Optional) Sets the router to log IS-IS adjacency state changes. Enter all to include all changes generated by events that are not related to the IS-IS hellos, including End System-to-Intermediate System PDUs and link state packets (LSPs).</p>
Step 15	<p>lsp-mtu <i>size</i></p> <p>Example:</p> <pre>Device(config-router)# lsp mtu 1560</pre>	<p>(Optional) Specifies the maximum LSP packet size, in bytes. The range is from 128 to 4352; the default is 1497 bytes.</p> <p>Note If a link in the network has a reduced MTU size, you must change the LSP MTU size on all the devices in the network.</p>
Step 16	<p>partition avoidance</p> <p>Example:</p>	<p>(Optional) Causes an IS-IS Level 1-2 border router to stop advertising the Level 1 area prefix into the Level 2</p>

	Command or Action	Purpose
	<code>Device(config-router)# partition avoidance</code>	backbone when full connectivity is lost among the border router, all adjacent level 1 routers, and end hosts.
Step 17	end Example: <code>Device(config)# end</code>	Returns to privileged EXEC mode.
Step 18	copy running-config startup-config Example: <code>Device# copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring IS-IS Interface Parameters

To configure IS-IS interface-specific parameters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>Device# configure terminal</code>	Enters global configuration mode.
Step 2	interface <i>interface-id</i> Example: <code>Device(config)# interface gigabitethernet 1/0/1</code>	Specifies the interface to be configured and enters interface configuration mode. If the interface is not already configured as a Layer 3 interface, enter the no switchport command to configure the interface into Layer 3 mode.
Step 3	isis metric <i>default-metric</i> [level-1 level-2] Example: <code>Device(config-if)# isis metric 15</code>	(Optional) Configures the metric (or cost) for the specified interface. The range is from 0 to 63; the default is 10. If no level is entered, the default is applied to both Level 1 and Level 2 routers.
Step 4	isis hello-interval {<i>seconds</i> minimal} [level-1 level-2] Example: <code>Device(config-if)# isis hello-interval minimal</code>	(Optional) Specifies the length of time between the hello packets sent by the device. By default, a value that is three times the hello interval <i>seconds</i> is advertised as the <i>holdtime</i> in the hello packets sent. With smaller hello intervals, topological changes are detected faster, but there is more routing traffic. <ul style="list-style-type: none"> • minimal—Causes the system to compute the hello interval based on the hello multiplier so that the resulting hold time is 1 second. • <i>seconds</i>—Range is from 1 to 65535; default is 10 seconds.

	Command or Action	Purpose
Step 5	isis hello-multiplier <i>multiplier</i> [level-1 level-2] Example: <pre>Device(config-if)# isis hello-multiplier 5</pre>	(Optional) Specifies the number of IS-IS hello packets a neighbor must miss before the device declares the adjacency as down. The range is from 3 to 1000; default is 3. Note Using a smaller hello multiplier causes fast convergence, but might result in routing instability.
Step 6	isis csnp-interval <i>seconds</i> [level-1 level-2] Example: <pre>Device(config-if)# isis csnp-interval 15</pre>	(Optional) Configures the IS-IS complete sequence number PDU (CSNP) interval for the interface. The range is from 0 to 65535; default is 10 seconds.
Step 7	isis retransmit-interval <i>seconds</i> Example: <pre>Device(config-if)# isis retransmit-interval 7</pre>	(Optional) Configures the number of seconds between the retransmission of IS-IS LSPs for point-to-point links. Specify an integer that is greater than the expected round-trip delay between any two routers on the network. The range is from 0 to 65535; default is 5 seconds.
Step 8	isis retransmit-throttle-interval <i>milliseconds</i> Example: <pre>Device(config-if)# isis retransmit-throttle-interval 4000</pre>	(Optional) Configures the IS-IS LSP retransmission throttle interval, which is the maximum rate (number of milliseconds between packets) at which IS-IS LSPs will be resent on point-to-point links. The range is from 0 to 65535; default is determined by the isis lsp-interval command.
Step 9	isis priority <i>value</i> [level-1 level-2] Example: <pre>Device(config-if)# isis priority 50</pre>	(Optional) Configures the priority for the designated router. The range is from 0 to 127; default is 64.
Step 10	isis circuit-type {level-1 level-1-2 level-2-only} Example: <pre>Device(config-if)# isis circuit-type level-1-2</pre>	(Optional) Configures the type of adjacency required for neighbors on the specified interface (specify the interface circuit type). <ul style="list-style-type: none"> • level-1—Level 1 adjacency is established if there is at least one area address that is common to both this node and its neighbors. • level-1-2—Level 1 and Level 2 adjacency are established if the neighbor is also configured as both Level 1 and Level 2, and there is at least one area in common. If there is no area in common, a Level 2 adjacency is established. This is the default option. • level 2—Level 2 adjacency is established. If the neighbor router is a Level 1 router, no adjacency is established.

	Command or Action	Purpose
Step 11	isis password <i>password</i> [level-1 level-2] Example: Device(config-if)# isis password secret	(Optional) Configures the authentication password for an interface. By default, authentication is disabled. Specifying Level 1 or Level 2 enables the password only for Level 1 or Level 2 routing, respectively. If you do not specify a level, the default is Level 1 and Level 2.
Step 12	end Example: Device(config)# end	Returns to privileged EXEC mode.

Monitoring and Maintaining IS-IS

You can display specific IS-IS statistics, such as the contents of routing tables, caches, and databases. You can also display information about specific interfaces, filters, or neighbors.

The following table lists the privileged EXEC commands for clearing and displaying IS-IS routing.

Table 17: IS-IS show Commands

Command
show ip route isis
show isis database
show isis routes
show isis spf-log
show isis topology
show route-map
trace clns destination

Feature Information for IS-IS

Table 18: Feature Information for IS-IS

Feature Name	Release	Feature Information
Intermediate System-to-Intermediate System (IS-IS)	Cisco IOS XE Fuji 16.9.2	This feature was introduced.



CHAPTER 8

Configuring Multi-VRF CE

- [Information About Multi-VRF CE, on page 115](#)
- [How to Configure Multi-VRF CE, on page 118](#)
- [Configuration Examples for Multi-VRF CE, on page 131](#)
- [Feature Information for Multi-VRF CE, on page 135](#)

Information About Multi-VRF CE

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over an ISP backbone network. A VPN is a collection of sites sharing a common routing table. A customer site is connected to the service-provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table, called a VPN routing/forwarding (VRF) table.

The switch supports multiple VPN routing/forwarding (multi-VRF) instances in customer edge (CE) devices (multi-VRF CE) when it is running the . Multi-VRF CE allows a service provider to support two or more VPNs with overlapping IP addresses.



Note The switch does not use Multiprotocol Label Switching (MPLS) to support VPNs.

Understanding Multi-VRF CE

Multi-VRF CE is a feature that allows a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. Multi-VRF CE uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but an interface cannot belong to more than one VRF at any time.



Note Multi-VRF CE interfaces must be Layer 3 interfaces.

Multi-VRF CE includes these devices:

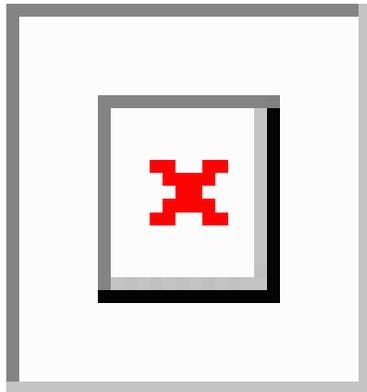
- Customer edge (CE) devices provide customers access to the service-provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the router and learns the remote VPN routes from it. A switch can be a CE.
- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as , RIPv2, OSPF, or EIGRP. The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service-provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF.
- Provider routers or core routers are any routers in the service provider network that do not attach to CE devices.

With multi-VRF CE, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. Multi-VRF CE extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

Network Topology

The figure shows a configuration using switches as multiple virtual CEs. This scenario is suited for customers who have low bandwidth requirements for their VPN service, for example, small companies. In this case, multi-VRF CE support is required in the switches. Because multi-VRF CE is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 5: Switches Acting as Multiple Virtual CEs



When the CE switch receives a command to add a Layer 3 interface to a VRF, it sets up the appropriate mapping between the VLAN ID and the policy label (PL) in multi-VRF-CE-related data structures and adds the VLAN ID and PL to the VLAN database.

When multi-VRF CE is configured, the Layer 3 forwarding table is conceptually partitioned into two sections:

- The multi-VRF CE routing section contains the routes from different VPNs.
- The global routing section contains routes to non-VPN networks, such as the Internet.

VLAN IDs from different VRFs are mapped into different policy labels, which are used to distinguish the VRFs during processing. For each new VPN route learned, the Layer 3 setup function retrieves the policy

label by using the VLAN ID of the ingress port and inserts the policy label and new route to the multi-VRF CE routing section. If the packet is received from a routed port, the port internal VLAN ID number is used; if the packet is received from an SVI, the VLAN number is used.

Packet-Forwarding Process

This is the packet-forwarding process in a multi-VRF-CE-enabled network:

- When the switch receives a packet from a VPN, the switch looks up the routing table based on the input policy label number. When a route is found, the switch forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. Then it performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.
- When a CE receives a packet from an egress PE, it uses the input policy label to look up the correct VPN routing table. If a route is found, it forwards the packet within the VPN.

Network Components

To configure VRF, you create a VRF table and specify the Layer 3 interface associated with the VRF. Then configure the routing protocols in the VPN and between the CE and the PE. The multi-VRF CE network has three major components:

- VPN route target communities—lists of all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- VPN forwarding—transports all traffic between all VPN community members across a VPN service-provider network.

VRF-Aware Services

IP services can be configured on global interfaces, and these services run within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-Aware services are implemented in platform-independent modules. VRF means multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

How to Configure Multi-VRF CE

Default Multi-VRF CE Configuration

Table 19: Default VRF Configuration

Feature	Default Setting
VRF	Disabled. No VRFs are defined.
Maps	No import maps, export maps, or route maps are defined.
VRF maximum routes	Fast Ethernet switches: 8000 Gigabit Ethernet switches: 12000.
Forwarding table	The default for an interface is the global routing table.

Multi-VRF CE Configuration Guidelines

- A switch with multi-VRF CE is shared by multiple customers, and each customer has its own routing table.
- Because customers use different VRF tables, the same IP addresses can be reused. Overlapped IP addresses are allowed in different VPNs.
- Multi-VRF CE lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. Each customer has its own VLAN.
- Multi-VRF CE does not support all MPLS-VRF functionality. It does not support label exchange, LDP adjacency, or labeled packets.
- For the PE router, there is no difference between using multi-VRF CE or using multiple CEs. In Figure 41-6, multiple virtual Layer 3 interfaces are connected to the multi-VRF CE device.
- The switch supports configuring VRF by using physical ports, VLAN SVIs, or a combination of both. The SVIs can be connected through an access port or a trunk port.
- A customer can use multiple VLANs as long as they do not overlap with those of other customers. A customer's VLANs are mapped to a specific routing table ID that is used to identify the appropriate routing tables stored on the switch.
- On the C9200L-24T-4G, C9200L-24P-4G, C9200L-48T-4G, C9200L-48P-4G, C9200L-24T-4X, C9200L-24P-4X, C9200L-48T-4X, and C9200L-48P-4X models of the Cisco Catalyst 9200 Series Switches, the switch supports one VRF.
- On the C9200-24T, C9200-24P, C9200-48T, and C9200-48P models of the Cisco Catalyst 9200 Series Switches, the switch supports 4 VRFs.
- Multi-VRF CE does not affect the packet switching rate.
- VPN multicast is not supported.
- You can enable VRF on a private VLAN, and the reverse.

- You cannot enable VRF when policy-based routing (PBR) is enabled on an interface, and the reverse.
- You cannot enable VRF when Web Cache Communication Protocol (WCCP) is enabled on an interface, and the reverse.

Configuring VRFs

Perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	ip routing Example: Device(config)# ip routing	Enables IP routing.
Step 3	ip vrf vrf-name Example: Device(config)# ip vrf vpn1	Names the VRF, and enter VRF configuration mode.
Step 4	rd route-distinguisher Example: Device(config-vrf)# rd 100:2	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y)
Step 5	route-target {export import both} route-target-ext-community Example: Device(config-vrf)# route-target both 100:2	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The <i>route-target-ext-community</i> should be the same as the <i>route-distinguisher</i> entered in Step 4.
Step 6	import map route-map Example: Device(config-vrf)# import map importmap1	(Optional) Associates a route map with the VRF.

	Command or Action	Purpose
Step 7	interface <i>interface-id</i> Example: <pre>Device(config-vrf)# interface gigabitethernet 1/0/1</pre>	Specifies the Layer 3 interface to be associated with the VRF, and enter interface configuration mode. The interface can be a routed port or SVI.
Step 8	ip vrf forwarding <i>vrf-name</i> Example: <pre>Device(config-if)# ip vrf forwarding vpn1</pre>	Associates the VRF with the Layer 3 interface. Note When ip vrf forwarding is enabled in the Management Interface, the access point does not join.
Step 9	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 10	show ip vrf [brief detail interfaces] [<i>vrf-name</i>] Example: <pre>Device# show ip vrf interfaces vpn1</pre>	Verifies the configuration. Displays information about the configured VRFs.
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring VRF-Aware Services

These services are VRF-Aware:

- ARP
- Ping
- Simple Network Management Protocol (SNMP)
- Unicast Reverse Path Forwarding (uRPF)
- Syslog
- Traceroute
- FTP and TFTP

Configuring VRF-Aware Services for ARP

Procedure

	Command or Action	Purpose
Step 1	show ip arp vrf <i>vrf-name</i> Example: Device# show ip arp vrf vpn1	Displays the ARP table in the specified VRF.

Configuring VRF-Aware Services for Ping

Procedure

	Command or Action	Purpose
Step 1	ping vrf <i>vrf-name</i> ip-host Example: Device# ping vrf vpn1 ip-host	Displays the ARP table in the specified VRF.

Configuring VRF-Aware Services for SNMP

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	snmp-server trap authentication vrf Example: Device(config)# snmp-server trap authentication vrf	Enables SNMP traps for packets on a VRF.
Step 3	snmp-server engineID remote <i>host vrf vpn-instance engine-id string</i> Example: Device(config)# snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100	Configures a name for the remote SNMP engine on a switch.
Step 4	snmp-server host <i>host vrf vpn-instance traps community</i> Example:	Specifies the recipient of an SNMP trap operation and specifies the VRF table to be used for sending SNMP traps.

	Command or Action	Purpose
	Device(config)# snmp-server host 172.16.20.3 vrf vpn1 traps comaccess	
Step 5	snmp-server host <i>host vrf vpn-instance informs community</i> Example: Device(config)# snmp-server host 172.16.20.3 vrf vpn1 informs comaccess	Specifies the recipient of an SNMP inform operation and specifies the VRF table to be used for sending SNMP informs.
Step 6	snmp-server user <i>user group remote host vrf vpn-instance security model</i> Example: Device(config)# snmp-server user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des	Adds a user to an SNMP group for a remote host on a VRF for SNMP access.
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring VRF-Aware Services for NTP

Configuring VRF-aware services for NTP comprises configuring the NTP servers and the NTP client interfaces connected to the NTP servers.

Before you begin

Ensure connectivity between the NTP client and servers. Configure a valid IP address and subnet on the client interfaces that are connected to the NTP servers.

Configuring VRF-Aware Services for NTP on NTP Client

Perform the following steps on the client interface that is connected to the NTP server.

Procedure

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

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	<code>interface interface-id</code> Example: Device(config)# <code>interface gigabitethernet 1/0/1</code>	Specifies the Layer 3 interface to be associated with the VRF, and enters the interface configuration mode.
Step 4	<code>vrf forwarding vrf-name</code> Example: Device(config-if)# <code>vrf forwarding A</code>	Associates the VRF with the Layer 3 interface.
Step 5	<code>ip address ip-address subnet-mask</code> Example: Device(config-if)# <code>ip address 1.1.1.1 255.255.255.0</code>	Enter the IP address for the interface.
Step 6	<code>no shutdown</code> Example: Device(config-if)# <code>no shutdown</code>	Enables the interface.
Step 7	<code>exit</code> Example: Device(config-if) <code>exit</code>	Exits the interface configuration mode.
Step 8	<code>ntp authentication-key number md5 md5-number</code> Example: Device(config)# <code>ntp authentication-key 1 md5 cisco123</code>	Defines the authentication keys. The device does not synchronize to a time source unless the source has one of these authentication keys and the key number is specified by the ntp trusted-key number command. Note The authentication key <i>number</i> and the MD5 <i>passwd</i> must be the same on both the client and server.
Step 9	<code>ntp authenticate</code> Example: Device(config)# <code>ntp authenticate</code>	Enables the NTP authentication feature. NTP authentication is disabled by default.
Step 10	<code>ntp trusted-key key-number</code> Example: Device(config)# <code>ntp trusted-key 1</code>	Specifies one or more keys that an NTP server must provide in its NTP packets in order for the NTP client to synchronize to it. The range for trusted keys is from 1 to 65535. This command provides protection against accidentally synchronizing the NTP client to an NTP server that is not trusted.
Step 11	<code>ntp server vrf vrf-name</code> Example: Device(config)# <code>ntp server vrf A 1.1.1.2 key 1</code>	Configures NTP Server in the specified VRF.

Configuring VRF-Aware Services for NTP on the NTP Server

Perform the following steps on the NTP server.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ntp authentication-key number md5 passowrd Example: Device(config)# ntp authentication-key 1 md5 cisco123	Defines the authentication keys. The device does not synchronize to a time source unless the source has one of these authentication keys and the key number is specified by the ntp trusted-key number command. <p>Note The authentication key <i>number</i> and the MD5 <i>passowrd</i> must be the same on both the client and server.</p>
Step 4	ntp authenticate Example: Device(config)# ntp authenticate	Enables the NTP authentication feature. NTP authentication is disabled by default.
Step 5	ntp trusted-key key-number Example: Device(config)# ntp trusted-key 1	Specifies one or more keys that an NTP server must provide in its NTP packets in order for the NTP client to synchronize to it. The range for trusted keys is from 1 to 65535. This command provides protection against accidentally synchronizing the NTP client to an NTP server that is not trusted.
Step 6	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/3	Specifies the Layer 3 interface to be associated with the VRF, and enters the interface configuration mode.
Step 7	vrf forwarding vrf-name Example: Device(config-if)# vrf forwarding A	Associates the VRF with the Layer 3 interface.
Step 8	ip address ip-address subnet-mask Example:	Enter the IP address for the interface.

	Command or Action	Purpose
	Device(config-if)# ip address 1.1.1.2 255.255.255.0	
Step 9	exit Example: Device(config-if) exit	Exits the interface configuration mode.

Configuring VRF-Aware Services for uRPF

uRPF can be configured on an interface assigned to a VRF, and source lookup is done in the VRF table.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 3	no switchport Example: Device(config-if)# no switchport	Removes the interface from Layer 2 configuration mode if it is a physical interface.
Step 4	ip vrf forwarding vrf-name Example: Device(config-if)# ip vrf forwarding vpn2	Configures VRF on the interface.
Step 5	ip address ip-address Example: Device(config-if)# ip address 10.1.5.1	Enters the IP address for the interface.
Step 6	ip verify unicast reverse-path Example: Device(config-if)# ip verify unicast reverse-path	Enables uRPF on the interface.
Step 7	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-if) # end	

Configuring VRF-Aware RADIUS

To configure VRF-Aware RADIUS, you must first enable AAA on a RADIUS server. The switch supports the **ip vrf forwarding** *vrf-name* server-group configuration and the **ip radius source-interface** global configuration commands, as described in the Per VRF AAA Feature Guide.

Configuring VRF-Aware Services for Syslog

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	logging on Example: Device(config) # logging on	Enables or temporarily disables logging of storage router event message.
Step 3	logging host ip-address vrf vrf-name Example: Device(config) # logging host 10.10.1.0 vrf vpn1	Specifies the host address of the syslog server where logging messages are to be sent.
Step 4	logging buffered logging buffered size debugging Example: Device(config) # logging buffered critical 6000 debugging	Logs messages to an internal buffer.
Step 5	logging trap debugging Example: Device(config) # logging trap debugging	Limits the logging messages sent to the syslog server.
Step 6	logging facility facility Example: Device(config) # logging facility user	Sends system logging messages to a logging facility.

	Command or Action	Purpose
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuring VRF-Aware Services for Traceroute

Procedure

	Command or Action	Purpose
Step 1	traceroute vrf <i>vrf-name</i> <i>ipaddress</i> Example: Device(config)# traceroute vrf vpn2 10.10.1.1	Specifies the name of a VPN VRF in which to find the destination address.

Configuring VRF-Aware Services for FTP and TFTP

So that FTP and TFTP are VRF-aware, you must configure some FTP/TFTP CLIs. For example, if you want to use a VRF table that is attached to an interface, say E1/0, you need to configure the `ip tftp source-interface E1/0` or the `ip ftp source-interface E1/0` command to inform TFTP or FTP server to use a specific routing table. In this example, the VRF table is used to look up the destination IP address. These changes are backward-compatible and do not affect existing behavior. That is, you can use the source-interface CLI to send packets out a particular interface even if no VRF is configured on that interface.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	ip ftp source-interface <i>interface-type interface-number</i> Example: Device(config)# ip ftp source-interface gigabitethernet 1/0/2	Specifies the source IP address for FTP connections.
Step 3	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 4	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 5	ip tftp source-interface <i>interface-type interface-number</i> Example: Device(config)# <code>ip tftp source-interface gigabitethernet 1/0/2</code>	Specifies the source IP address for TFTP connections.
Step 6	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

Configuring Multicast VRFs

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	ip routing Example: Device(config)# <code>ip routing</code>	Enables IP routing mode.
Step 3	ip vrf <i>vrf-name</i> Example: Device(config)# <code>ip vrf vpn1</code>	Names the VRF, and enter VRF configuration mode.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# <code>rd 100:2</code>	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y)
Step 5	route-target { export import both } <i>route-target-ext-community</i>	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an

	Command or Action	Purpose
	Example: Device(config-vrf)# route-target import 100:2	IP address and an arbitrary number (A.B.C.D:y). The <i>route-target-ext-community</i> should be the same as the <i>route-distinguisher</i> entered in Step 4.
Step 6	import map <i>route-map</i> Example: Device(config-vrf)# import map importmap1	(Optional) Associates a route map with the VRF.
Step 7	ip multicast-routing vrf <i>vrf-name</i> distributed Example: Device(config-vrf)# ip multicast-routing vrf vpn1 distributed	(Optional) Enables global multicast routing for VRF table.
Step 8	interface <i>interface-id</i> Example: Device(config-vrf)# interface gigabitethernet 1/0/2	Specifies the Layer 3 interface to be associated with the VRF, and enter interface configuration mode. The interface can be a routed port or an SVI.
Step 9	ip vrf forwarding <i>vrf-name</i> Example: Device(config-if)# ip vrf forwarding vpn1	Associates the VRF with the Layer 3 interface.
Step 10	ip address <i>ip-address</i> mask Example: Device(config-if)# ip address 10.1.5.1 255.255.255.0	Configures IP address for the Layer 3 interface.
Step 11	ip pim sparse-dense mode Example: Device(config-if)# ip pim sparse-dense mode	Enables PIM on the VRF-associated Layer 3 interface.
Step 12	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 13	show ip vrf [brief detail interfaces] [<i>vrf-name</i>] Example: Device# show ip vrf detail vpn1	Verifies the configuration. Displays information about the configured VRFs.

	Command or Action	Purpose
Step 14	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring a VPN Routing Session

Routing within the VPN can be configured with any supported routing protocol (RIP, OSPF, EIGRP, or) or with static routing. The configuration shown here is for OSPF, but the process is the same for other protocols.



Note To configure an EIGRP routing process to run within a VRF instance, you must configure an autonomous-system number by entering the **autonomous-system** *autonomous-system-number* address-family configuration mode command.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	router ospf process-id vrf vrf-name Example: Device(config)# <code>router ospf 1 vrf vpn1</code>	Enables OSPF routing, specifies a VPN forwarding table, and enter router configuration mode.
Step 3	log-adjacency-changes Example: Device(config-router)# <code>log-adjacency-changes</code>	(Optional) Logs changes in the adjacency state. This is the default state.
Step 4	redistribute isissubnets Example: Device(config-router)# <code>redistribute isis subnets</code>	Sets the switch to redistribute information from the ISIS network to the OSPF network.
Step 5	network network-number area area-id Example: Device(config-router)# <code>network 1 area 2</code>	Defines a network address and mask on which OSPF runs and the area ID for that network address.

	Command or Action	Purpose
Step 6	end Example: Device(config-router)# end	Returns to privileged EXEC mode.
Step 7	show ip ospf process-id Example: Device# show ip ospf 1	Verifies the configuration of the OSPF network.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring Multi-VRF CE

Table 20: Commands for Displaying Multi-VRF CE Information

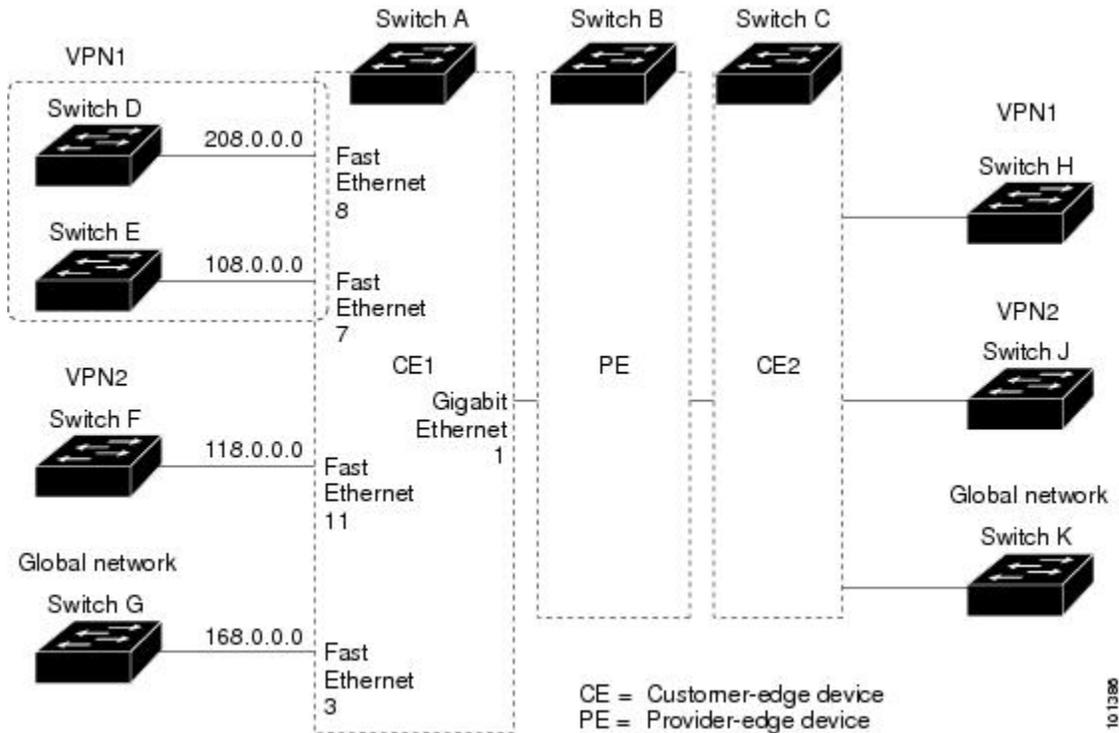
show ip protocols vrf vrf-name	Displays routing protocol information as a VRF.
show ip route vrf vrf-name [connected] [protocol [as-number]] [list] [mobile] [odr] [profile] [static] [summary] [supernets-only]	Displays IP routing table information as a VRF.
show ip vrf [brief detail interfaces] [vrf-name]	Displays information about the defined VRFs.

Configuration Examples for Multi-VRF CE

Multi-VRF CE Configuration Example

OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The examples following the illustration show how to configure a switch as CE Switch A, and the VRF configuration for customer switches D and F. Commands for configuring CE Switch C and the other customer switches are not included but would be similar.

Figure 6: Multi-VRF CE Configuration Example



On Switch A, enable routing and configure VRF.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# ip vrf v11
Device(config-vrf)# rd 800:1
Device(config-vrf)# route-target export 800:1
Device(config-vrf)# route-target import 800:1
Device(config-vrf)# exit
Device(config)# ip vrf v12
Device(config-vrf)# rd 800:2
Device(config-vrf)# route-target export 800:2
Device(config-vrf)# route-target import 800:2
Device(config-vrf)# exit
```

Configure the loopback and physical interfaces on Switch A. Gigabit Ethernet port 1 is a trunk connection to the PE. Gigabit Ethernet ports 8 and 11 connect to VPNs:

```
Device(config)# interface loopback1
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 8.8.1.8 255.255.255.0
Device(config-if)# exit

Device(config)# interface loopback2
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 8.8.2.8 255.255.255.0
Device(config-if)# exit

Device(config)# interface gigabitethernet1/0/5
Device(config-if)# switchport trunk encapsulation dot1q
```

```

Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/8
Device(config-if)# switchport access vlan 208
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit

```

Configure the VLANs used on Switch A. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for the VPNs that include Switch F and Switch D, respectively:

```

Device(config)# interface vlan10
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 38.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan20
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 83.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan118
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 118.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan208
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 208.0.0.8 255.255.255.0
Device(config-if)# exit

```

Configure OSPF routing in VPN1 and VPN2.

```

Device(config)# router ospf 1 vrf v11
Device(config-router)# redistribute isis subnets
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# exit
Device(config)# router ospf 2 vrf v12
Device(config-router)# redistribute isis subnets
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# exit

```

Switch D belongs to VPN 1. Configure the connection to Switch A by using these commands.

```

Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 208.0.0.20 255.255.255.0
Device(config-if)# exit

Device(config)# router ospf 101
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# end

```

Switch F belongs to VPN 2. Configure the connection to Switch A by using these commands.

```

Device# configure terminal

```

```

Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit

Device(config)# interface vlan118
Device(config-if)# ip address 118.0.0.11 255.255.255.0
Device(config-if)# exit

Device(config)# router ospf 101
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# end

```

When used on switch B (the PE router), these commands configure only the connections to the CE device, Switch A.

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit
Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf v1
Router(config-router-af)# neighbor 38.0.0.8 remote-as 800
Router(config-router-af)# neighbor 38.0.0.8 activate

```

```
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

Feature Information for Multi-VRF CE

Table 21: Feature Information for Multi-VRF CE

Feature Name	Release	Feature Information
Multi-VRF CE	Cisco IOS XE Fuji 16.9.2	This feature was introduced



CHAPTER 9

Configuring Unicast Reverse Path Forwarding

- [Configuring Unicast Reverse Path Forwarding, on page 137](#)

Configuring Unicast Reverse Path Forwarding

The unicast reverse path forwarding (unicast RPF) feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.



Note

- Unicast RPF is supported in .
-



CHAPTER 10

Protocol-Independent Features

- [Protocol-Independent Features, on page 139](#)

Protocol-Independent Features

This section describes IP routing protocol-independent features that are available on switches running the feature set .

Distributed Cisco Express Forwarding

Information About Cisco Express Forwarding

Cisco Express Forwarding (CEF) is a Layer 3 IP switching technology used to optimize network performance. CEF implements an advanced IP look-up and forwarding algorithm to deliver maximum Layer 3 switching performance. CEF is less CPU-intensive than fast switching route caching, allowing more CPU processing power to be dedicated to packet forwarding. In a switch stack, the hardware uses distributed CEF (dCEF) in the stack. In dynamic networks, fast switching cache entries are frequently invalidated because of routing changes, which can cause traffic to be process switched using the routing table, instead of fast switched using the route cache. CEF and dCEF use the Forwarding Information Base (FIB) lookup table to perform destination-based switching of IP packets.

The two main components in CEF and dCEF are the distributed FIB and the distributed adjacency tables.

- The FIB is similar to a routing table or information base and maintains a mirror image of the forwarding information in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next-hop address information based on the information in the IP routing table. Because the FIB contains all known routes that exist in the routing table, CEF eliminates route cache maintenance, is more efficient for switching traffic, and is not affected by traffic patterns.
- Nodes in the network are said to be adjacent if they can reach each other with a single hop across a link layer. CEF uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Because the switch or switch stack uses Application Specific Integrated Circuits (ASICs) to achieve Gigabit-speed line rate IP traffic, CEF or dCEF forwarding applies only to the software-forwarding path, that is, traffic that is forwarded by the CPU.

How to Configure Cisco Express Forwarding

CEF or distributed CEF is enabled globally by default. If for some reason it is disabled, you can re-enable it by using the **ip cef** or **ip cef distributed** global configuration command.

The default configuration is CEF or dCEF enabled on all Layer 3 interfaces. Entering the **no ip route-cache cef** interface configuration command disables CEF for traffic that is being forwarded by software. This command does not affect the hardware forwarding path. Disabling CEF and using the **debug ip packet detail** privileged EXEC command can be useful to debug software-forwarded traffic. To enable CEF on an interface for the software-forwarding path, use the **ip route-cache cef** interface configuration command.



Caution Although the **no ip route-cache cef** interface configuration command to disable CEF on an interface is visible in the CLI, we strongly recommend that you do not disable CEF or dCEF on interfaces except for debugging purposes.

To enable CEF or dCEF globally and on an interface for software-forwarded traffic if it has been disabled:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	ip cef Example: Device(config)# <code>ip cef</code>	Enables CEF operation on a non-stacking switch. Go to Step 4.
Step 3	ip cef distributed Example: Device(config)# <code>ip cef distributed</code>	Enables CEF operation on a active switch.
Step 4	interface <i>interface-id</i> Example: Device(config)# <code>interface gigabitethernet 1/0/1</code>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 5	ip route-cache cef Example: Device(config-if)# <code>ip route-cache cef</code>	Enables CEF on the interface for software-forwarded traffic.
Step 6	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-if)# end	
Step 7	show ip cef Example: Device# show ip cef	Displays the CEF status on all interfaces.
Step 8	show cef linecard [detail] Example: Device# show cef linecard detail	(Optional) Displays CEF-related interface information on a non-stacking switch.
Step 9	show cef linecard [slot-number] [detail] Example: Device# show cef linecard 5 detail	(Optional) Displays CEF-related interface information on a switch by stack member for all switches in the stack or for the specified switch. (Optional) For <i>slot-number</i> , enter the stack member switch number.
Step 10	show cef interface [interface-id] Example: Device# show cef interface gigabitethernet 1/0/1	Displays detailed CEF information for all interfaces or the specified interface.
Step 11	show adjacency Example: Device# show adjacency	Displays CEF adjacency table information.
Step 12	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Load-Balancing Scheme for CEF Traffic

Restrictions for Configuring a Load-Balancing Scheme for CEF Traffic

- You must globally configure load balancing on device or device stack members in the same way.
- Per-packet load balancing for CEF traffic is not supported.

CEF Load-Balancing Overview

CEF load balancing allows you to optimize resources by distributing traffic over multiple paths. CEF load balancing works based on a combination of source and destination packet information.

You can configure load balancing on a per-destination. Because load-balancing decisions are made on the outbound interface, load balancing must be configured on the outbound interface.

Per-Destination Load Balancing for CEF Traffic

Per-destination load balancing allows the device to use multiple paths to achieve load sharing across multiple source-destination host pairs. Packets for a given source-destination host pair are guaranteed to take the same path, even if multiple paths are available. Traffic streams destined for different pairs tend to take different paths.

Per-destination load balancing is enabled by default when you enable CEF. To use per-destination load balancing, you do not perform any additional tasks once CEF is enabled. Per-destination is the load-balancing method of choice for most situations.

Because per-destination load balancing depends on the statistical distribution of traffic, load sharing becomes more effective as the number of source-destination host pairs increases.

You can use per-destination load balancing to ensure that packets for a given host pair arrive in order. All packets intended for a certain host pair are routed over the same link (or links).

Load-Balancing Algorithms for CEF Traffic

The following load-balancing algorithms are provided for use with CEF traffic. Select a load-balancing algorithm with the **ip cef load-sharing algorithm** command.

- **Original algorithm**—The original load-balancing algorithm produces distortions in load sharing across multiple devices because the same algorithm was used on every device. Depending on your network environment, you should select the algorithm.
- **Universal algorithm**—The universal load-balancing algorithm allows each device on the network to make a different load sharing decision for each source-destination address pair, which resolves load-sharing imbalances. The device is set to perform universal load sharing by default.

How to Configure a Load-Balancing for CEF Traffic

The following sections provide information on configuring load-balancing for CEF traffic.

Enabling or Disabling CEF Per-Destination Load Balancing

To enable or disable CEF per-destination load balancing, perform the following procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **[no] ip load-sharing per-destination**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device# enable	Enters global configuration mode.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config-if)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	[no] ip load-sharing per-destination Example: Device(config-if)# ip load-sharing per-destination	Enables per-destination load balancing for CEF on the interface. The no ip load-sharing per-destination command disables per-destination load balancing for CEF on the interface.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Selecting a Tunnel Load-Balancing Algorithm for CEF Traffic

Select the tunnel algorithm when your network environment contains only a few source and destination pairs. The device is set to perform universal load sharing by default.

To select a tunnel load-balancing algorithm for CEF traffic, perform the following procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef load-sharing algorithm** {**original** | **universal** [*id*] }
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device# enable	Enters global configuration mode.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip cef load-sharing algorithm {original universal [id]} Example: Device(config)# ip cef load-sharing algorithm universal	Selects a CEF load-balancing algorithm. <ul style="list-style-type: none"> • The original keyword sets the load-balancing algorithm to the original algorithm, based on a source IP and destination IP hash. • The universal keyword sets the load-balancing algorithm to one that uses a source IP, destination IP, Layer 3 Protocol, Layer 4 source port, Layer 4 destination port and IPv6 flow label (for IPv6 traffic). • The <i>id</i> argument is a fixed identifier.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuration Examples for CEF Traffic Load-Balancing

The following sections provide configuration examples for CEF traffic load-balancing.

Example: Enabling or Disabling CEF Per-Destination Load Balancing

Per-destination load balancing is enabled by default when you enable CEF. The following example shows how to disable per-destination load balancing:

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# no ip load-sharing per-destination
Device(config-if)# end
```

Number of Equal-Cost Routing Paths

Information About Equal-Cost Routing Paths

When a router has two or more routes to the same network with the same metrics, these routes can be thought of as having an equal cost. The term parallel path is another way to see occurrences of equal-cost routes in a routing table. If a router has two or more equal-cost paths to a network, it can use them concurrently. Parallel paths provide redundancy in case of a circuit failure and also enable a router to load balance packets over the available paths for more efficient use of available bandwidth. Equal-cost routes are supported across switches in a stack.

Even though the router automatically learns about and configures equal-cost routes, you can control the maximum number of parallel paths supported by an IP routing protocol in its routing table. Although the switch software allows a maximum of 32 equal-cost routes, the switch hardware will never use more than 16 paths per route.

How to Configure Equal-Cost Routing Paths

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router {rip ospf eigrp} Example: Device(config)# router eigrp	Enters router configuration mode.
Step 3	maximum-paths <i>maximum</i> Example: Device(config-router)# maximum-paths 2	Sets the maximum number of parallel paths for the protocol routing table. The range is from 1 to 16; the default is 4 for most IP routing protocols, but only 1 for BGP.
Step 4	end Example: Device(config-router)# end	Returns to privileged EXEC mode.
Step 5	show ip protocols Example: Device# show ip protocols	Verifies the setting in the <i>Maximum path</i> field.
Step 6	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# copy running-config startup-config	

Static Unicast Routes

Information About Static Unicast Routes

Static unicast routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot build a route to a particular destination and are useful for specifying a gateway of last resort to which all unroutable packets are sent.

The switch retains static routes until you remove them. However, you can override static routes with dynamic routing information by assigning administrative distance values. Each dynamic routing protocol has a default administrative distance, as listed in Table 41-16. If you want a static route to be overridden by information from a dynamic routing protocol, set the administrative distance of the static route higher than that of the dynamic protocol.

Table 22: Dynamic Routing Protocol Default Administrative Distances

Route Source	Default Distance
Connected interface	0
Static route	1
Enhanced IRGP summary route	5
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
Unknown	225

Static routes that point to an interface are advertised through RIP, IGRP, and other dynamic routing protocols, whether or not static **redistribute** router configuration commands were specified for those routing protocols. These static routes are advertised because static routes that point to an interface are considered in the routing table to be connected and hence lose their static nature. However, if you define a static route to an interface that is not one of the networks defined in a network command, no dynamic routing protocols advertise the route unless a **redistribute** static command is specified for these protocols.

When an interface goes down, all static routes through that interface are removed from the IP routing table. When the software can no longer find a valid next hop for the address specified as the forwarding router's address in a static route, the static route is also removed from the IP routing table.

Configuring Static Unicast Routes

Static unicast routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot build a route to a particular destination and are useful for specifying a gateway of last resort to which all unroutable packets are sent.

Follow these steps to configure a static route:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip route prefix mask <i>{address interface}</i> [<i>distance</i>] Example: Device(config)# ip route prefix mask gigabitethernet 1/0/4	Establish a static route.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 5	show ip route Example: Device# show ip route	Displays the current state of the routing table to verify the configuration.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

What to do next

Use the **no ip route prefix mask** *{address| interface}* global configuration command to remove a static route. The device retains static routes until you remove them.

Default Routes and Networks

Information About Default Routes and Networks

A router might not be able to learn the routes to all other networks. To provide complete routing capability, you can use some routers as smart routers and give the remaining routers default routes to the smart router. (Smart routers have routing table information for the entire internetwork.) These default routes can be dynamically learned or can be configured in the individual routers. Most dynamic interior routing protocols include a mechanism for causing a smart router to generate dynamic default information that is then forwarded to other routers.

If a router has a directly connected interface to the specified default network, the dynamic routing protocols running on that device generate a default route. In RIP, it advertises the pseudonetwork 0.0.0.0.

A router that is generating the default for a network also might need a default of its own. One way a router can generate its own default is to specify a static route to the network 0.0.0.0 through the appropriate device.

When default information is passed through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the optimal default network as its default route. In IGRP networks, there might be several candidate networks for the system default. Cisco routers use administrative distance and metric information to set the default route or the gateway of last resort.

If dynamic default information is not being passed to the system, candidates for the default route are specified with the **ip default-network** global configuration command. If this network appears in the routing table from any source, it is flagged as a possible choice for the default route. If the router has no interface on the default network, but does have a path to it, the network is considered as a possible candidate, and the gateway to the best default path becomes the gateway of last resort.

How to Configure Default Routes and Networks

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	ip default-network <i>network number</i> Example: Device(config)# <code>ip default-network 1</code>	Specifies a default network.
Step 3	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 4	show ip route Example:	Displays the selected default route in the gateway of last resort display.

	Command or Action	Purpose
	Device# show ip route	
Step 5	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Route Maps to Redistribute Routing Information

Information About Route Maps

The switch can run multiple routing protocols simultaneously, and it can redistribute information from one routing protocol to another. Redistributing information from one routing protocol to another applies to all supported IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by defining enhanced packet filters or route maps between the two domains. The **match** and **set** route-map configuration commands define the condition portion of a route map. The **match** command specifies that a criterion must be matched. The **set** command specifies an action to be taken if the routing update meets the conditions defined by the match command. Although redistribution is a protocol-independent feature, some of the **match** and **set** route-map configuration commands are specific to a particular protocol.

One or more **match** commands and one or more **set** commands follow a **route-map** command. If there are no **match** commands, everything matches. If there are no **set** commands, nothing is done, other than the match. Therefore, you need at least one **match** or **set** command.



Note A route map with no **set** route-map configuration commands is sent to the CPU, which causes high CPU utilization.

You can also identify route-map statements as **permit** or **deny**. If the statement is marked as a deny, the packets meeting the match criteria are sent back through the normal forwarding channels (destination-based routing). If the statement is marked as permit, set clauses are applied to packets meeting the match criteria. Packets that do not meet the match criteria are forwarded through the normal routing channel.

How to Configure a Route Map

Although each of Steps 3 through 14 in the following section is optional, you must enter at least one **match** route-map configuration command and one **set** route-map configuration command.



Note The keywords are the same as defined in the procedure to control the route distribution.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	route-map <i>map-tag</i> [permit deny] [<i>sequence number</i>] Example: Device(config)# route-map rip-to-ospf permit 4	Defines any route maps used to control redistribution and enter route-map configuration mode. <i>map-tag</i> —A meaningful name for the route map. The redistribute router configuration command uses this name to reference this route map. Multiple route maps might share the same map tag name. (Optional) If permit is specified and the match criteria are met for this route map, the route is redistributed as controlled by the set actions. If deny is specified, the route is not redistributed. <i>sequence number</i> (Optional)— Number that indicates the position a new route map is to have in the list of route maps already configured with the same name.
Step 3	match as-path <i>path-list-number</i> Example: Device(config-route-map)#match as-path 10	Matches a BGP AS path access list.
Step 4	match community-list <i>community-list-number</i> [exact] Example: Device(config-route-map)# match community-list 150	Matches a BGP community list.
Step 5	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [<i>...access-list-number</i> <i>...access-list-name</i>] Example: Device(config-route-map)# match ip address 5 80	Matches a standard access list by specifying the name or number. It can be an integer from 1 to 199.
Step 6	match metric <i>metric-value</i> Example: Device(config-route-map)# match metric 2000	Matches the specified route metric. The <i>metric-value</i> can be an EIGRP metric with a specified value from 0 to 4294967295.
Step 7	match ip next-hop { <i>access-list-number</i> <i>access-list-name</i> } [<i>...access-list-number</i> <i>...access-list-name</i>] Example:	Matches a next-hop router address passed by one of the access lists specified (numbered from 1 to 199).

	Command or Action	Purpose
	Device(config-route-map)# match ip next-hop 8 45	
Step 8	match tag tag value [...tag-value] Example: Device(config-route-map)# match tag 3500	Matches the specified tag value in a list of one or more route tag values. Each can be an integer from 0 to 4294967295.
Step 9	match interfacetype number [...type-number] Example: Device(config-route-map)# match interface gigabitethernet 1/0/1	Matches the specified next hop route out one of the specified interfaces.
Step 10	match ip route-source {access-list-number access-list-name} [...access-list-number ...access-list-name] Example: Device(config-route-map)# match ip route-source 10 30	Matches the address specified by the specified advertised access lists.
Step 11	match route-type {local internal external [type-1 type-2]} Example: Device(config-route-map)# match route-type local	Matches the specified route-type : <ul style="list-style-type: none"> • local—Locally generated BGP routes. • internal—OSPF intra-area and interarea routes or EIGRP internal routes. • external—OSPF external routes (Type 1 or Type 2) or EIGRP external routes.
Step 12	set dampening halflife reuse suppress max-suppress-time Example: Device(config-route-map)# set dampening 30 1500 10000 120	Sets BGP route dampening factors.
Step 13	set local-preference value Example: Device(config-route-map)# set local-preference 100	Assigns a value to a local BGP path.
Step 14	set origin {igp egp as incomplete} Example: Device(config-route-map)#set origin igp	Sets the BGP origin code.

	Command or Action	Purpose
Step 15	set as-path {tag prepend <i>as-path-string</i> } Example: <pre>Device(config-route-map)# set as-path tag</pre>	Modifies the BGP autonomous system path.
Step 16	set level {level-1 level-2 level-1-2 stub-area backbone} Example: <pre>Device(config-route-map)# set level level-1-2</pre>	Sets the level for routes that are advertised into the specified area of the routing domain. The stub-area and backbone are OSPF NSSA and backbone areas.
Step 17	set metric <i>metric value</i> Example: <pre>Device(config-route-map)# set metric 100</pre>	Sets the metric value to give the redistributed routes (for EIGRP only). The <i>metric value</i> is an integer from -294967295 to 294967295.
Step 18	set metric <i>bandwidth delay reliability loading mtu</i> Example: <pre>Device(config-route-map)# set metric 10000 10 255 1 1500</pre>	Sets the metric value to give the redistributed routes (for EIGRP only): <ul style="list-style-type: none"> • <i>bandwidth</i>—Metric value or IGRP bandwidth of the route in kilobits per second in the range 0 to 4294967295 • <i>delay</i>—Route delay in tens of microseconds in the range 0 to 4294967295. • <i>reliability</i>—Likelihood of successful packet transmission expressed as a number between 0 and 255, where 255 means 100 percent reliability and 0 means no reliability. • <i>loading</i>—Effective bandwidth of the route expressed as a number from 0 to 255 (255 is 100 percent loading). • <i>mtu</i>—Minimum maximum transmission unit (MTU) size of the route in bytes in the range 0 to 4294967295.
Step 19	set metric-type {type-1 type-2} Example: <pre>Device(config-route-map)# set metric-type type-2</pre>	Sets the OSPF external metric type for redistributed routes.
Step 20	set metric-type internal Example: <pre>Device(config-route-map)# set metric-type internal</pre>	Sets the multi-exit discriminator (MED) value on prefixes advertised to external BGP neighbor to match the IGP metric of the next hop.

	Command or Action	Purpose
Step 21	set weight <i>number</i> Example: Device(config-route-map)# set weight 100	Sets the BGP weight for the routing table. The value can be from 1 to 65535.
Step 22	end Example: Device(config-route-map)# end	Returns to privileged EXEC mode.
Step 23	show route-map Example: Device# show route-map	Displays all route maps configured or only the one specified to verify configuration.
Step 24	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

How to Control Route Distribution

Although each of Steps 3 through 14 in the following section is optional, you must enter at least one **match** route-map configuration command and one **set** route-map configuration command.



Note The keywords are the same as defined in the procedure to configure the route map for redistribution.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is a hop count, and the IGRP metric is a combination of five qualities. In these situations, an artificial metric is assigned to the redistributed route. Uncontrolled exchanging of routing information between different routing protocols can create routing loops and seriously degrade network operation.

If you have not defined a default redistribution metric that replaces metric conversion, some automatic metric translations occur between routing protocols:

- RIP can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- Any protocol can redistribute other routing protocols if a default mode is in effect.

Procedure

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

	Command or Action	Purpose
Step 2	router { rip ospf eigrp } Example: Device(config)# router eigrp 10	Enters router configuration mode.
Step 3	redistribute <i>protocol</i> [<i>process-id</i>] { level-1 level-1-2 level-2 } [metric <i>metric-value</i>] [metric-type <i>type-value</i>] [match internal external <i>type-value</i>] [tag <i>tag-value</i>] [route-map <i>map-tag</i>] [weight <i>weight</i>] [subnets] Example: Device(config-router)# redistribute eigrp 1	Redistributes routes from one routing protocol to another routing protocol. If no route-maps are specified, all routes are redistributed. If the keyword route-map is specified with no <i>map-tag</i> , no routes are distributed.
Step 4	default-metric <i>number</i> Example: Device(config-router)# default-metric 1024	Cause the current routing protocol to use the same metric value for all redistributed routes (RIP and OSPF).
Step 5	default-metric bandwidth delay reliability loading mtu Example: Device(config-router)# default-metric 1000 100 250 100 1500	Cause the EIGRP routing protocol to use the same metric value for all non-EIGRP redistributed routes.
Step 6	end Example: Device(config-router)# end	Returns to privileged EXEC mode.
Step 7	show route-map Example: Device# show route-map	Displays all route maps configured or only the one specified to verify configuration.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Policy-Based Routing

Restrictions for Configuring PBR

- Policy-based routing (PBR) is not supported to forward traffic into GRE tunnel. This applies to PBR applied on any interface and forwarding traffic into GRE tunnel (by means of PBR next-hop or default next-hop or set interface).

- PBR is not supported on GRE tunnel itself (applied under the GRE tunnel itself).

Information About Policy-Based Routing

You can use policy-based routing (PBR) to configure a defined policy for traffic flows. By using PBR, you can have more control over routing by reducing the reliance on routes derived from routing protocols. PBR can specify and implement routing policies that allow or deny paths based on:

- Identity of a particular end system
- Application
- Protocol

You can use PBR to provide equal-access and source-sensitive routing, routing based on interactive versus batch traffic, or routing based on dedicated links. For example, you could transfer stock records to a corporate office on a high-bandwidth, high-cost link for a short time while transmitting routine application data such as e-mail over a low-bandwidth, low-cost link.

With PBR, you classify traffic using access control lists (ACLs) and then make traffic go through a different path. PBR is applied to incoming packets. All packets received on an interface with PBR enabled are passed through route maps. Based on the criteria defined in the route maps, packets are forwarded (routed) to the appropriate next hop.

- Route map statement marked as permit is processed as follows:
 - A match command can match on length or multiple ACLs. A route map statement can contain multiple match commands. Logical or algorithm function is performed across all the match commands to reach a permit or deny decision.

For example:

```
match length A B
match ip address acl1 acl2
match ip address acl3
```

A packet is permitted if it is permitted by match length A B or acl1 or acl2 or acl3

- If the decision reached is permit, then the action specified by the set command is applied on the packet .
 - If the decision reached is deny, then the PBR action (specified in the set command) is not applied. Instead the processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.
- For PBR, route-map statements marked as deny are not supported.

You can use standard IP ACLs to specify match criteria for a source address or extended IP ACLs to specify match criteria based on an application, a protocol type, or an end station. The process proceeds through the route map until a match is found. If no match is found, normal destination-based routing occurs. There is an implicit deny at the end of the list of match statements.

If match clauses are satisfied, you can use a set clause to specify the IP addresses identifying the next hop router in the path.

How to Configure PBR

- Multicast traffic is not policy-routed. PBR applies only to unicast traffic.
- You can enable PBR on a routed port or an SVI.
- The switch supports PBR based on match length.
- You can apply a policy route map to an EtherChannel port channel in Layer 3 mode, but you cannot apply a policy route map to a physical interface that is a member of the EtherChannel. If you try to do so, the command is rejected. When a policy route map is applied to a physical interface, that interface cannot become a member of an EtherChannel.
- You can define a maximum of 128 IP policy route maps on the switch or switch stack.
- You can define a maximum of 512 access control entries (ACEs) for PBR on the switch or switch stack.
- When configuring match criteria in a route map, follow these guidelines:
 - Do not match ACLs that permit packets destined for a local address.
- VRF and PBR are mutually exclusive on a switch interface. You cannot enable VRF when PBR is enabled on an interface. The reverse is also true, you cannot enable PBR when VRF is enabled on an interface.
- Web Cache Communication Protocol (WCCP) and PBR are mutually exclusive on a switch interface. You cannot enable WCCP when PBR is enabled on an interface. The reverse is also true, you cannot enable PBR when WCCP is enabled on an interface.
- The number of hardware entries used by PBR depends on the route map itself, the ACLs used, and the order of the ACLs and route-map entries.
- PBR based on TOS, DSCP and IP Precedence are not supported.
- Set interface, set default next-hop and set default interface are not supported.
- **ip next-hop recursive** and **ip next-hop verify availability** features are not available and the next-hop should be directly connected.
- Policy-maps with no set actions are supported. Matching packets are routed normally.
- Policy-maps with no match clauses are supported. Set actions are applied to all packets.

By default, PBR is disabled on the switch. To enable PBR, you must create a route map that specifies the match criteria and the resulting action. Then, you must enable PBR for that route map on an interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

Packets that are generated by the switch (CPU), or local packets, are not normally policy-routed. When you globally enable local PBR on the switch, all unicast packets that originate on the switch are subject to local PBR. The protocols that are supported for local PBR are NTP, DNS, MSDP, SYSLOG and TFTP. Local PBR is disabled by default.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> [permit] [<i>sequence number</i>] Example: Device(config)# route-map pbr-map permit	Defines route maps that are used to control where packets are output, and enters route-map configuration mode. <ul style="list-style-type: none"> • <i>map-tag</i> – A meaningful name for the route map. The ip policy route-map interface configuration command uses this name to reference the route map. Multiple route-map statements with the same map tag define a single route map. • (Optional) permit – If permit is specified and the match criteria are met for this route map, the route is policy routed as defined by the set actions. • (Optional) <i>sequence number</i> – The sequence number shows the position of the route-map statement in the given route map.
Step 4	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [<i>access-list-number</i> ... <i>access-list-name</i>] Example: Device(config-route-map)# match ip address 110 140	Matches the source and destination IP addresses that are permitted by one or more standard or extended access lists. ACLs can match on more than one source and destination IP address. If you do not specify a match command, the route map is applicable to all packets.
Step 5	match length min max Example: Device(config-route-map)# match length 64 1500	Matches the length of the packet.
Step 6	set ip next-hop ip-address [... <i>ip-address</i>] Example: Device(config-route-map)# set ip next-hop 10.1.6.2	Specifies the action to be taken on the packets that match the criteria. Sets next hop to which to route the packet (the next hop must be adjacent).
Step 7	exit Example: Device(config-route-map)# exit	Returns to global configuration mode.
Step 8	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to be configured.

	Command or Action	Purpose
Step 9	ip policy route-map <i>map-tag</i> Example: Device(config-if)# ip policy route-map pbr-map	Enables PBR on a Layer 3 interface, and identify the route map to use. You can configure only one route map on an interface. However, you can have multiple route map entries with different sequence numbers. These entries are evaluated in the order of sequence number until the first match. If there is no match, packets are routed as usual.
Step 10	ip route-cache policy Example: Device(config-if)# ip route-cache policy	(Optional) Enables fast-switching PBR. You must enable PBR before enabling fast-switching PBR.
Step 11	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 12	ip local policy route-map <i>map-tag</i> Example: Device(config)# ip local policy route-map local-pbr	(Optional) Enables local PBR to perform policy-based routing on packets originating at the switch. This applies to packets generated by the switch, and not to incoming packets.
Step 13	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 14	show route-map [<i>map-name</i>] Example: Device# show route-map	(Optional) Displays all the route maps configured or only the one specified to verify configuration.
Step 15	show ip policy Example: Device# show ip policy	(Optional) Displays policy route maps attached to the interface.
Step 16	show ip local policy Example: Device# show ip local policy	(Optional) Displays whether or not local policy routing is enabled and, if so, the route map being used.

Filtering Routing Information

You can filter routing protocol information by performing the tasks described in this section.



Note When routes are redistributed between OSPF processes, no OSPF metrics are preserved.

Setting Passive Interfaces

To prevent other routers on a local network from dynamically learning about routes, you can use the **passive-interface** router configuration command to keep routing update messages from being sent through a router interface. When you use this command in the OSPF protocol, the interface address you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified router interface.

In networks with many interfaces, to avoid having to manually set them as passive, you can set all interfaces to be passive by default by using the **passive-interface default** router configuration command and manually setting interfaces where adjacencies are desired.

Use a network monitoring privileged EXEC command such as **show ip ospf interface** to verify the interfaces that you enabled as passive, or use the **show ip interface** privileged EXEC command to verify the interfaces that you enabled as active.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router { rip ospf eigrp } Example: Device(config)# router ospf	Enters router configuration mode.
Step 3	passive-interface interface-id Example: Device(config-router)# passive-interface gigabitethernet 1/0/1	Suppresses sending routing updates through the specified Layer 3 interface.
Step 4	passive-interface default Example: Device(config-router)# passive-interface default	(Optional) Sets all interfaces as passive by default.
Step 5	no passive-interface interface type Example: Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5	(Optional) Activates only those interfaces that need to have adjacencies sent.
Step 6	network network-address Example: Device(config-router)# network 10.1.1.1	(Optional) Specifies the list of networks for the routing process. The <i>network-address</i> is an IP address.

	Command or Action	Purpose
Step 7	end Example: Device(config-router)# end	Returns to privileged EXEC mode.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Controlling Advertising and Processing in Routing Updates

You can use the **distribute-list** router configuration command with access control lists to suppress routes from being advertised in routing updates and to prevent other routers from learning one or more routes. When used in OSPF, this feature applies to only external routes, and you cannot specify an interface name.

You can also use a **distribute-list** router configuration command to avoid processing certain routes listed in incoming updates. (This feature does not apply to OSPF.)

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	router { rip eigrp } Example: Device(config)# router eigrp 10	Enters router configuration mode.
Step 3	distribute-list {access-list-number access-list-name} out [interface-name routing process autonomous-system-number] Example: Device(config-router)# distribute 120 out gigabitethernet 1/0/7	Permits or denies routes from being advertised in routing updates, depending upon the action listed in the access list.
Step 4	distribute-list {access-list-number access-list-name} in [type-number] Example: Device(config-router)# distribute-list 125 in	Suppresses processing in routes listed in updates.

	Command or Action	Purpose
Step 5	end Example: <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Filtering Sources of Routing Information

Because some routing information might be more accurate than others, you can use filtering to prioritize information coming from different sources. An administrative distance is a rating of the trustworthiness of a routing information source, such as a router or group of routers. In a large network, some routing protocols can be more reliable than others. By specifying administrative distance values, you enable the router to intelligently discriminate between sources of routing information. The router always picks the route whose routing protocol has the lowest administrative distance.

Because each network has its own requirements, there are no general guidelines for assigning administrative distances.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	router { rip ospf eigrp } Example: <pre>Device(config)# router eigrp 10</pre>	Enters router configuration mode.
Step 3	distance weight {ip-address {ip-address mask}} [ip access list] Example: <pre>Device(config-router)# distance 50 10.1.5.1</pre>	Defines an administrative distance. <i>weight</i> —The administrative distance as an integer from 10 to 255. Used alone, <i>weight</i> specifies a default administrative distance that is used when no other specification exists for a routing information source. Routes with a distance of 255 are not installed in the routing table. (Optional) <i>ip access list</i> —An IP standard or extended access list to be applied to incoming routing updates.
Step 4	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-router)# end	
Step 5	show ip protocols Example: Device# show ip protocols	Displays the default administrative distance for a specified routing process.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Managing Authentication Keys

Key management is a method of controlling authentication keys used by routing protocols. Not all protocols can use key management. Authentication keys are available for EIGRP and RIP Version 2.

Prerequisites

Before you manage authentication keys, you must enable authentication. See the appropriate protocol section to see how to enable authentication for that protocol. To manage authentication keys, define a key chain, identify the keys that belong to the key chain, and specify how long each key is valid. Each key has its own key identifier (specified with the **key number** key chain configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and Message Digest 5 (MD5) authentication key in use.

How to Configure Authentication Keys

You can configure multiple keys with life times. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest, and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. Note that the router must know these lifetimes.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	key chain <i>name-of-chain</i> Example: Device(config)# key chain key10	Identifies a key chain, and enter key chain configuration mode.

	Command or Action	Purpose
Step 3	<p>key number</p> <p>Example:</p> <pre>Device(config-keychain)# key 2000</pre>	Identifies the key number. The range is 0 to 2147483647.
Step 4	<p>key-string text</p> <p>Example:</p> <pre>Device(config-keychain)# Room 20, 10th floor</pre>	Identifies the key string. The string can contain from 1 to 80 uppercase and lowercase alphanumeric characters, but the first character cannot be a number.
Step 5	<p>accept-lifetime start-time {infinite end-time duration seconds}</p> <p>Example:</p> <pre>Device(config-keychain)# accept-lifetime 12:30:00 Jan 25 1009 infinite</pre>	<p>(Optional) Specifies the time period during which the key can be received.</p> <p>The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i>. The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite.</p>
Step 6	<p>send-lifetime start-time {infinite end-time duration seconds}</p> <p>Example:</p> <pre>Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite</pre>	<p>(Optional) Specifies the time period during which the key can be sent.</p> <p>The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i>. The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite.</p>
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-keychain)# end</pre>	Returns to privileged EXEC mode.
Step 8	<p>show key chain</p> <p>Example:</p> <pre>Device# show key chain</pre>	Displays authentication key information.
Step 9	<p>copy running-config startup-config</p> <p>Example:</p> <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.



CHAPTER 11

Configuring Generic Routing Encapsulation(GRE) Tunnel IP Source and Destination VRF Membership

- [Restrictions for GRE Tunnel IP Source and Destination VRF Membership, on page 165](#)
- [Information About GRE Tunnel IP Source and Destination VRF Membership, on page 165](#)
- [How to Configure GRE Tunnel IP Source and Destination VRF Membership, on page 166](#)
- [Configuration Example for GRE Tunnel IP Source and Destination VRF Membership, on page 167](#)
- [Additional References, on page 168](#)
- [Feature History for Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership, on page 169](#)

Restrictions for GRE Tunnel IP Source and Destination VRF Membership

- Both ends of the tunnel must reside within the same VRF.
- The VRF associated with the tunnel vrf command is the same as the VRF associated with the physical interface over which the tunnel sends packets (outer IP packet routing).
- The VRF associated with the tunnel by using the ip vrf forwarding command is the VRF that the packets are to be forwarded in as the packets exit the tunnel (inner IP packet routing).
- The feature does not support the fragmentation of multicast packets passing through a multicast tunnel.
- The feature does not support the ISIS (Intermediate System to intermediate system) protocol.

Information About GRE Tunnel IP Source and Destination VRF Membership

This feature allows you to configure the source and destination of a tunnel to belong to any Virtual Private Network (VPN) routing and forwarding (VRF) table. A VRF table stores routing data for each VPN. The

VRF table defines the VPN membership of a customer site attached to the network access server (NAS). Each VRF table comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

Previously, GRE IP tunnels required the IP tunnel destination to be in the global routing table. The implementation of this feature allows you to configure a tunnel source and destination to belong to any VRF. As with existing GRE tunnels, the tunnel becomes disabled if no route to the tunnel destination is defined.

How to Configure GRE Tunnel IP Source and Destination VRF Membership

Follow these steps to configure GRE Tunnel IP Source and Destination VRF Membership:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel***number*
4. **ip vrf forwarding***vrf-name*
5. **ip address***ip-address subnet-mask*
6. **tunnel source** { *ip-address* | *type number* }
7. **tunnel destination** { *hostname* | *ip-address* }
8. **tunnel vrf***vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 0	Enters interface configuration mode for the specified interface. <ul style="list-style-type: none">• <i>number</i> is the number associated with the tunnel interface.
Step 4	ip vrf forwarding <i>vrf-name</i> Example: Device(config-if)# ip vrf forwarding green	Associates a virtual private network (VPN) routing and forwarding (VRF) instance with an interface or subinterface. <ul style="list-style-type: none">• <i>vrf-name</i> is the name assigned to a VRF.
Step 5	ip address <i>ip-address subnet-mask</i>	Specifies the interface IP address and subnet mask.

	Command or Action	Purpose
	Example: Device(config-if)# ip address 10.7.7.7 255.255.255.255	<ul style="list-style-type: none"> • ip-address specifies the IP address of the interface. • subnet-mask specifies the subnet mask of the interface.
Step 6	tunnel source { <i>ip-address</i> <i>type number</i> } Example: Device(config-if)# tunnel source loop 0	Specifies the source of the tunnel interface. <ul style="list-style-type: none"> • ip-address specifies the IP address to use as the source address for packets in the tunnel. • type specifies the interface type (for example, serial). • number specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when added to a system, and can be displayed using the show interfaces command.
Step 7	tunnel destination { <i>hostname</i> <i>ip-address</i> } Example: Device(config-if)# tunnel destination 10.5.5.5	Defines the tunnel destination. <ul style="list-style-type: none"> • hostname specifies the name of the host destination. • ip-address specifies the IP address of the host destination.
Step 8	tunnel vrf <i>vrf-name</i> Example: Device(config-if)# tunnel vrf finance1	Associates a VPN routing and forwarding (VRF) instance with a specific tunnel destination. <ul style="list-style-type: none"> • vrf-name is the name assigned to a VRF.

Configuration Example for GRE Tunnel IP Source and Destination VRF Membership

In this example, packets received on interface e0 using VRF green are forwarded out of the tunnel through interface e1 using VRF blue.

```
ip vrf blue rd 1:1

ip vrf green rd 1:2

interface loop0
ip vrf forwarding blue
ip address 10.7.7.7 255.255.255.255

interface tunnel0
ip vrf forwarding green
ip address 10.3.3.3 255.255.255.0 tunnel source loop 0
tunnel destination 10.5.5.5 tunnel vrf blue

interface ethernet0
ip vrf forwarding green
```

```

ip address 10.1.1.1 255.255.255.0

interface ethernet1
ip vrf forwarding blue
ip address 10.2.2.2 255.255.255.0

ip route vrf blue 10.5.5.5 255.255.255.0 ethernet 1

```

Additional References

Table 23: Related Documents

Related Topic	Document Title
VRF tables	"Configuring Multiprotocol Label Switching" chapter of the Cisco IOS Switching Services Configuration Guide, Release 12.2
Tunnels	Cisco IOS Interface Configuration Guide, Release 12.2

Table 24: Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature	--

Table 25: RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Table 26: Related DoTechnical Assistancecuments

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature History for Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership

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 <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Table 27: Feature History for Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership

Feature Name	Releases	Feature Information
Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership	Cisco IOS 16.6.1	The Generic Routing Encapsulation Tunnel IP Source and Destination VRF Membership feature allows you to configure the source and destination of a tunnel to belong to any virtual private network (VPN) routing and forwarding (VRF) table.

