



Routing Configuration Guide, Cisco IOS XE Everest 16.5.1a (Catalyst 9500 Switches)

First Published: 2017-06-20

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

Finding Feature Information 1

Prerequisites for Bidirectional Forwarding Detection 1

Restrictions for Bidirectional Forwarding Detection 2

Information About Bidirectional Forwarding Detection 2

BFD Operation 2

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 Static Routing 17

Configuring BFD Echo Mode 19

Creating and Configuring BFD Templates 20

Monitoring and Troubleshooting BFD 21

Feature Information for Bidirectional Forwarding Detection 22

CHAPTER 2

Configuring MSDP 23

Finding Feature Information 23

Information About Configuring MSDP 23

MSDP Overview 24

MSDP Operation 24

MSDP Benefits 25

How to Configure MSDP 26

Default MSDP Configuration 26

Configuring a Default MSDP Peer 26

Caching Source-Active State	28
Controlling Source Information that Your Switch Originates	30
Redistributing Sources	30
Filtering Source-Active Request Messages	32
Controlling Source Information that Your Switch Forwards	34
Using a Filter	34
Using TTL to Limit the Multicast Data Sent in SA Messages	36
Controlling Source Information that Your Switch Receives	37
Configuring an MSDP Mesh Group	39
Shutting Down an MSDP Peer	41
Including a Bordering PIM Dense-Mode Region in MSDP	42
Configuring an Originating Address other than the RP Address	43
Monitoring and Maintaining MSDP	44
Configuration Examples for Configuring MSDP	45
Configuring a Default MSDP Peer: Example	45
Caching Source-Active State: Example	46
Controlling Source Information that Your Switch Originates: Example	46
Controlling Source Information that Your Switch Forwards: Example	46
Controlling Source Information that Your Switch Receives: Example	46
Feature Information for Multicast Source Discovery Protocol	47

CHAPTER 3
Configuring IP Unicast Routing 49

Restrictions for IP Unicast Routing	49
Information About Configuring IP Unicast Routing	49
Information About IP Routing	50
Types of Routing	50
Classless Routing	51
Address Resolution	53
Proxy ARP	54
ICMP Router Discovery Protocol	54
UDP Broadcast Packets and Protocols	54
Broadcast Packet Handling	55
IP Broadcast Flooding	55
How to Configure IP Routing	56

How to Configure IP Addressing	57
Default IP Addressing Configuration	57
Assigning IP Addresses to Network Interfaces	58
Using Subnet Zero	60
Disabling Classless Routing	61
Configuring Address Resolution Methods	61
Defining a Static ARP Cache	62
Setting ARP Encapsulation	63
Enabling Proxy ARP	64
Routing Assistance When IP Routing is Disabled	65
Proxy ARP	66
Default Gateway	66
ICMP Router Discovery Protocol (IRDP)	67
Configuring Broadcast Packet Handling	69
Enabling Directed Broadcast-to-Physical Broadcast Translation	69
Forwarding UDP Broadcast Packets and Protocols	71
Establishing an IP Broadcast Address	72
Flooding IP Broadcasts	73
Monitoring and Maintaining IP Addressing	75
How to Configure IP Unicast Routing	76
Enabling IP Unicast Routing	76
Example of Enabling IP Routing	77
What to Do Next	77
Monitoring and Maintaining the IP Network	77
Feature Information for IP Unicast Routing	77
 CHAPTER 4	
Protocol-Independent Features	79
Protocol-Independent Features	79
Distributed Cisco Express Forwarding	79
Information About Cisco Express Forwarding	79
How to Configure Cisco Express Forwarding	80
Load-Balancing Scheme for CEF Traffic	81
Restrictions for Configuring a Load-Balancing Scheme for CEF Traffic	81
CEF Load-Balancing Overview	82

Per-Destination Load Balancing for CEF Traffic	82
Load-Balancing Algorithms for CEF Traffic	82
How to Configure a Load-Balancing for CEF Traffic	82
Configuration Examples for CEF Traffic Load-Balancing	84
Number of Equal-Cost Routing Paths	84
Information About Equal-Cost Routing Paths	84
How to Configure Equal-Cost Routing Paths	85
Static Unicast Routes	85
Information About Static Unicast Routes	85
Configuring Static Unicast Routes	86
Default Routes and Networks	87
Information About Default Routes and Networks	87
How to Configure Default Routes and Networks	88
Route Maps to Redistribute Routing Information	89
Information About Route Maps	89
How to Configure a Route Map	89
How to Control Route Distribution	93
Policy-Based Routing	95
Restrictions for Configuring PBR	95
Information About Policy-Based Routing	95
How to Configure PBR	96
Filtering Routing Information	99
Setting Passive Interfaces	99
Controlling Advertising and Processing in Routing Updates	100
Filtering Sources of Routing Information	101
Managing Authentication Keys	102
Prerequisites	103
How to Configure Authentication Keys	103

CHAPTER 5
Implementing Multiprotocol BGP for IPv6 105

Information About Implementing Multiprotocol BGP for IPv6	105
Multiprotocol BGP Extensions for IPv6	105
IPv6 Multiprotocol BGP Peering Using a Link-Local Address	105
Multiprotocol BGP for the IPv6 Multicast Address Family	105

Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family	106
How to Implement Multiprotocol BGP for IPv6	106
Configuring an IPv6 BGP Routing Process and BGP Router ID	106
Configuring IPv6 Multiprotocol BGP Between Two Peers	108
Configuring IPv6 Multiprotocol BGP Between Two Peers Using Link-Local Addresses	109
Troubleshooting Tips	112
Configuring an IPv6 Multiprotocol BGP Peer Group	112
Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes	114
Redistributing Prefixes into IPv6 Multiprotocol BGP	116
Advertising Routes into IPv6 Multiprotocol BGP	117
Advertising IPv4 Routes Between IPv6 BGP Peers	119
Assigning BGP Administrative Distance for Multicast BGP Routes	121
Generating IPv6 Multicast BGP Updates	122
Configuring the IPv6 BGP Graceful Restart Capability	123
Resetting IPv6 BGP Sessions	124
Verifying the IPv6 Multiprotocol BGP Configuration	125
Configuration Examples for Implementing Multiprotocol BGP for IPv6	126
Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer	126
Example: Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address	126
Example Configuring an IPv6 Multiprotocol BGP Peer Group	127
Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes	127
Example Redistributing Prefixes into IPv6 Multiprotocol BGP	128
Example: Advertising Routes into IPv6 Multiprotocol BGP	128
Example: Advertising IPv4 Routes Between IPv6 Peers	128
Additional References for Implementing Multiprotocol BGP for IPv6	129
Feature Information for Implementing Multiprotocol BGP for IPv6	129



CHAPTER 1

Configuring Bidirectional Forwarding Detection

- [Bidirectional Forwarding Detection, on page 1](#)

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 reconvergence time will be consistent and predictable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

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 support is not available starting Cisco Denali IOS XE 16.3.1

Information About Bidirectional Forwarding Detection

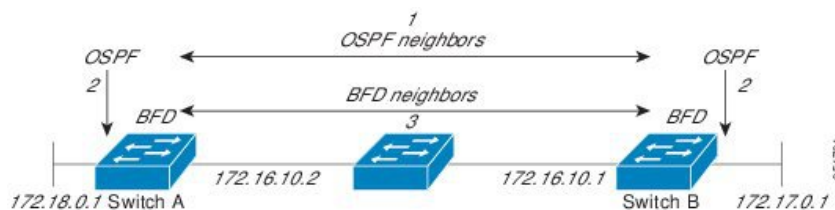
BFD Operation

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.

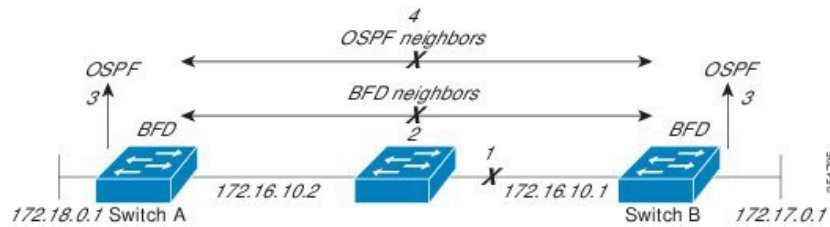
BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers. Therefore, in order for a BFD session 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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	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: <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 4	bfd interval <i>milliseconds</i> min_rx <i>milliseconds</i> multiplier <i>interval-multiplier</i> Example: <pre>Device(config-if)# bfd interval 100 min_rx 100 multiplier 3</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

	Command or Action	Purpose
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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-tag</i> Example: <pre>Device(config)# router bgp tag1</pre>	Specifies a BGP process and enters router configuration mode.
Step 4	neighbor <i>ip-address</i> fall-over bfd Example: <pre>Device(config-router)# neighbor 172.16.10.2 fall-over bfd</pre>	Enables BFD support for fallover.

	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 detail</pre>	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip bgp neighbor Example: <pre>Device# show ip bgp neighbor</pre>	(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.

Procedure

	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 eigrp as-number Example: Device(config)# router eigrp 123	Configures the EIGRP routing process and enters router configuration mode.
Step 4	Do one of the following: • bfd all-interfaces • bfd interface type number Example: Device(config-router)# bfd all-interfaces Example: Device(config-router)# bfd interface GigabitFastEthernet 1/0/1	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.
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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router isis area-tag Example: <pre>Device(config)# router isis tag1</pre>	Specifies an IS-IS process and enters router configuration mode.
Step 4	bfd all-interfaces Example: <pre>Device(config-router)# bfd all-interfaces</pre>	Enables BFD globally on all interfaces associated with the IS-IS routing process.
Step 5	exit Example: <pre>Device(config-router)# exit</pre>	(Optional) Returns the router to global configuration mode.

	Command or Action	Purpose
Step 6	interface <i>type number</i> Example: <pre>Device(config)# interface fastethernet 6/0</pre>	(Optional) Enters interface configuration mode.
Step 7	ip router isis [<i>tag</i>] Example: <pre>Device(config-if)# ip router isis tag1</pre>	(Optional) Enables support for IPv4 routing on the interface.
Step 8	isis bfd [disable] Example: <pre>Device(config-if)# isis bfd</pre>	(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: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 10	show bfd neighbors [details] Example: <pre>Device# show bfd neighbors details</pre>	(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: <pre>Device# show clns interface</pre>	(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.

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enables privileged EXEC mode. <ul style="list-style-type: none"> • 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	interface type number Example: Device(config)# interface fastethernet 6/0	Enters interface configuration mode.
Step 4	ip router isis [tag] 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. 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.
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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router ospf <i>process-id</i> Example: <pre>Device(config)# router ospf 4</pre>	Specifies an OSPF process and enters router configuration mode.
Step 4	bfd all-interfaces Example:	Enables BFD globally on all interfaces associated with the OSPF routing process.

	Command or Action	Purpose
	Device(config-router)# bfd all-interfaces	
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: Device# show ip ospf	(Optional) Displays information that can help verify if BFD for OSPF has been enabled.

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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: <pre>Device(config)# interface fastethernet 6/0</pre>	Enters interface configuration mode.
Step 4	ip ospf bfd [disable] Example: <pre>Device(config-if)# ip ospf bfd</pre>	Enables or 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 5	end Example: <pre>Device(config-if)# end</pre>	Exits interface 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) 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: <pre>Device# show ip ospf</pre>	(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 reenable 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.

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: 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.

	Command or Action	Purpose
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: Device(config)# standby bfd all-interfaces	(Optional) Enables HSRP support for BFD on 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.

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: Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Example:	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface serial 2/0	
Step 4	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • ip address <i>ipv4-address mask</i> • ipv6 address <i>ipv6-address/mask</i> <p>Example:</p> <p>Configuring an IPv4 address for the interface:</p> <pre>Device(config-if)# ip address 10.201.201.1 255.255.255.0</pre> <p>Configuring an IPv6 address for the interface:</p> <pre>Device(config-if)# ipv6 address 2001:db8:1:1::1/32</pre>	Configures an IP address for the interface.
Step 5	<p>bfd interval <i>milliseconds min_rx milliseconds multiplier interval-multiplier</i></p> <p>Example:</p> <pre>Device(config-if)# bfd interval 500 min_rx 500 multiplier 5</pre>	<p>Enables BFD on the interface.</p> <p>The bfd interval configuration is removed when the subinterface on which it is configured is removed.</p> <p>The bfd interval configuration is not removed when:</p> <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	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 7	<p>ip route static bfd <i>interface-type interface-number ip-address [group group-name [passive]]</i></p> <p>Example:</p>	<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.

	Command or Action	Purpose
	<pre>Device(config)# ip route static bfd TenGigabitEthernet1/0/1 10.10.10.2 group group1 passive</pre>	
Step 8	<p>ip route <i>[vrf vrf-name] prefix mask {ip-address interface-type interface-number [ip-address]} [dhcp] [distance] [name next-hop-name] [permanent track number] [tag tag]</i></p> <p>Example:</p> <pre>Device(config)# ip route 10.0.0.0 255.0.0.0</pre>	Specifies a static route BFD neighbor.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	<p>show ip static route</p> <p>Example:</p> <pre>Device# show ip static route</pre>	(Optional) Displays static route database information.
Step 11	<p>show ip static route bfd</p> <p>Example:</p> <pre>Device# show ip static route bfd</pre>	(Optional) Displays information about the static BFD configuration from the configured BFD groups and nongroup entries.
Step 12	<p>exit</p> <p>Example:</p> <pre>Device# exit</pre>	Exits privileged EXEC mode and returns to user EXEC mode.

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.

Procedure

	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	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	no bfd echo Example: <pre>Router(config)# no bfd echo</pre>	Disables BFD echo mode. <ul style="list-style-type: none"> • Use the no form to disable BFD echo mode.
Step 4	end Example: <pre>Router(config)# end</pre>	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.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	bfd-template single-hop <i>template-name</i> Example: <pre>Device(config)# bfd-template single-hop bfdtemplate1</pre>	Creates a single-hop BFD template and enters BFD configuration mode.
Step 4	interval min-tx <i>milliseconds</i> min-rx <i>milliseconds</i> multiplier <i>multiplier-value</i> 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.

Procedure

	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 Everest 16.5.1a	This feature was introduced



CHAPTER 2

Configuring MSDP

- [Finding Feature Information, on page 23](#)
- [Information About Configuring MSDP, on page 23](#)
- [How to Configure MSDP, on page 26](#)
- [Monitoring and Maintaining MSDP, on page 44](#)
- [Configuration Examples for Configuring MSDP, on page 45](#)
- [Feature Information for Multicast Source Discovery Protocol, on page 47](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Information About Configuring MSDP

This section describes how to configure the Multicast Source Discovery Protocol (MSDP) on the switch. The MSDP connects multiple Protocol-Independent Multicast sparse-mode (PIM-SM) domains.

MSDP is not fully supported in this software release because of a lack of support for Multicast Border Gateway Protocol (MBGP), which works closely with MSDP. However, it is possible to create default peers that MSDP can operate with if MBGP is not running.



Note To use this feature, the active switch must be running the Network Advantage feature set.

MSDP Overview

MSDP allows multicast sources for a group to be known to all rendezvous points (RPs) in different domains. Each PIM-SM domain uses its own RPs and does not depend on RPs in other domains. An RP runs MSDP over the Transmission Control Protocol (TCP) to discover multicast sources in other domains.

An RP in a PIM-SM domain has an MSDP peering relationship with MSDP-enabled devices in another domain. The peering relationship occurs over a TCP connection, primarily exchanging a list of sources sending to multicast groups. The TCP connections between RPs are achieved by the underlying routing system. The receiving RP uses the source lists to establish a source path.

The purpose of this topology is to have domains discover multicast sources in other domains. If the multicast sources are of interest to a domain that has receivers, multicast data is delivered over the normal, source-tree building mechanism in PIM-SM. MSDP is also used to announce sources sending to a group. These announcements must originate at the domain's RP.

MSDP depends heavily on the Border Gateway Protocol (BGP) or MBGP for interdomain operation. We recommend that you run MSDP in RPs in your domain that are RPs for sources sending to global groups to be announced to the Internet.

MSDP Operation

When a source sends its first multicast packet, the first-hop router (*designated router* or RP) directly connected to the source sends a PIM register message to the RP. The RP uses the register message to register the active source and to forward the multicast packet down the shared tree in the local domain. With MSDP configured, the RP also forwards a source-active (SA) message to all MSDP peers. The SA message identifies the source, the group the source is sending to, and the address of the RP or the originator ID (the IP address of the interface used as the RP address), if configured.

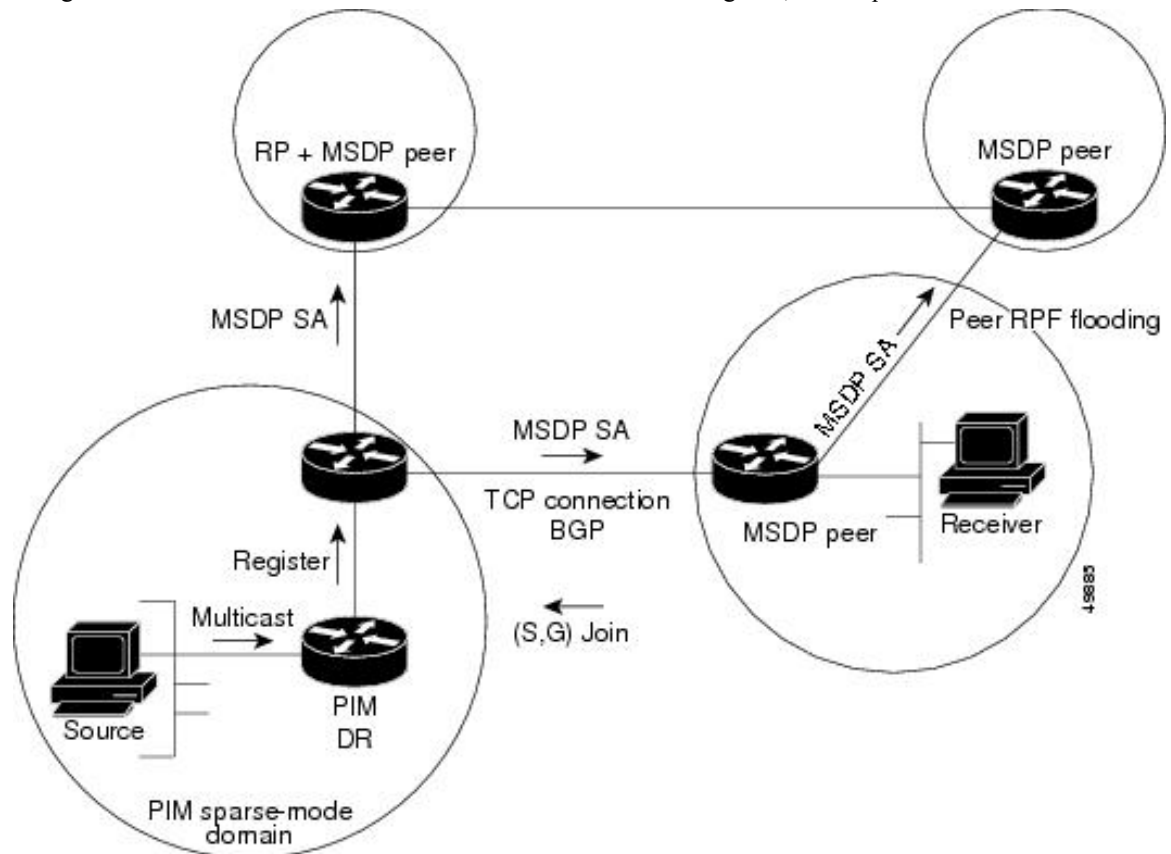
Each MSDP peer receives and forwards the SA message away from the originating RP to achieve peer reverse-path flooding (RPF). The MSDP device examines the BGP or MBGP routing table to discover which peer is the next hop toward the originating RP of the SA message. Such a peer is called an *RPF peer* (reverse-path forwarding peer). The MSDP device forwards the message to all MSDP peers other than the RPF peer. For information on how to configure an MSDP peer when BGP and MBGP are not supported, see the [Configuring a Default MSDP Peer, on page 26](#).

If the MSDP peer receives the same SA message from a non-RPF peer toward the originating RP, it drops the message. Otherwise, it forwards the message to all its MSDP peers.

The RP for a domain receives the SA message from an MSDP peer. If the RP has any join requests for the group the SA message describes and if the (*,G) entry exists with a nonempty outgoing interface list, the domain is interested in the group, and the RP triggers an (S,G) join toward the source. After the (S,G) join reaches the source's DR, a branch of the source tree has been built from the source to the RP in the remote domain. Multicast traffic can now flow from the source across the source tree to the RP and then down the shared tree in the remote domain to the receiver.

Figure 1: MSDP Running Between RP Peers

This figure shows MSDP operating between two MSDP peers. PIM uses MSDP as the standard mechanism to register a source with the RP of a domain. When MSDP is configured, this sequence occurs.



By default, the switch does not cache source or group pairs from received SA messages. When the switch forwards the MSDP SA information, it does not store it in memory. Therefore, if a member joins a group soon after an SA message is received by the local RP, that member needs to wait until the next SA message to hear about the source. This delay is known as join latency.

Local RPs can send SA requests and get immediate responses for all active sources for a given group. By default, the switch does not send any SA request messages to its MSDP peers when a new member joins a group and wants to receive multicast traffic. The new member waits to receive the next periodic SA message.

If you want a new member of a group to learn the active multicast sources in a connected PIM sparse-mode domain that are sending to a group, configure the switch to send SA request messages to the specified MSDP peer when a new member joins a group.

MSDP Benefits

MSDP has these benefits:

- It breaks up the shared multicast distribution tree. You can make the shared tree local to your domain. Your local members join the local tree, and join messages for the shared tree never need to leave your domain.

- PIM sparse-mode domains can rely only on their own RPs, decreasing reliance on RPs in another domain. This increases security because you can prevent your sources from being known outside your domain.
- Domains with only receivers can receive data without globally advertising group membership.
- Global source multicast routing table state is not required, saving memory.

How to Configure MSDP

Default MSDP Configuration

MSDP is not enabled, and no default MSDP peer exists.

Configuring a Default MSDP Peer

Before you begin

Configure an MSDP peer.

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 msdp default-peer <i>ip-address</i> <i>name</i> [prefix-list <i>list</i>] Example: Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a	Defines a default peer from which to accept all MSDP SA messages. <ul style="list-style-type: none"> • For <i>ip-address</i> / <i>name</i>, enter the IP address or Domain Name System (DNS) server name of the MSDP default peer. • (Optional) For prefix-list <i>list</i>, enter the list name that specifies the peer to be the default peer only for the listed prefixes. You can have multiple active default peers when you have a prefix list associated with each.

	Command or Action	Purpose
		<p>When you enter multiple ip msdp default-peer commands with the prefix-list keyword, you use all the default peers at the same time for different RP prefixes. This syntax is typically used in a service provider cloud that connects stub site clouds.</p> <p>When you enter multiple ip msdp default-peer commands without the prefix-list keyword, a single active peer accepts all SA messages. If that peer fails, the next configured default peer accepts all SA messages. This syntax is typically used at a stub site.</p>
Step 4	<p>ip prefix-list <i>name</i> [<i>description string</i>] seq number {permit deny} <i>network length</i></p> <p>Example:</p> <pre>Router(config)# ip prefix-list site-a seq 3 permit 12 network length 128</pre>	<p>(Optional) Creates a prefix list using the name specified in Step 2.</p> <ul style="list-style-type: none"> • (Optional) For description string, enter a description of up to 80 characters to describe this prefix list. • For seq number, enter the sequence number of the entry. The range is 1 to 4294967294. • The deny keyword denies access to matching conditions. • The permit keyword permits access to matching conditions. • For <i>network length</i>, specify the network number and length (in bits) of the network mask that is permitted or denied.
Step 5	<p>ip msdp description {<i>peer-name</i> <i>peer-address</i>} <i>text</i></p> <p>Example:</p> <pre>Router(config)# ip msdp description peer-name site-b</pre>	<p>(Optional) Configures a description for the specified peer to make it easier to identify in a configuration or in show command output.</p> <p>By default, no description is associated with an MSDP peer.</p>
Step 6	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	show running-config Example: Device# show running-config	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.

Caching Source-Active State

If you want to sacrifice some memory in exchange for reducing the latency of the source information, you can configure the Device to cache SA messages. Perform the following steps to enable the caching of source/group pairs:

Follow these steps to enable the caching of source/group pairs:

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 msdp cache-sa-state [list access-list-number] Example: Device(config)# ip msdp cache-sa-state 100	Enables the caching of source/group pairs (create an SA state). Those pairs that pass the access list are cached. For list access-list-number , the range is 100 to 199.

	Command or Action	Purpose
		Note An alternative to this command is the ip msdp sa-reques global configuration command, which causes the Device to send an SA request message to the MSDP peer when a new member for a group becomes active.
Step 4	<p>access-list <i>access-list-number</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i></p> <p>Example:</p> <pre>Device(config)# access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255</pre>	<p>Creates an IP extended access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, the range is 100 to 199. Enter the same number created in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>protocol</i>, enter ip as the protocol name. For <i>source</i>, enter the number of the network or host from which the packet is being sent. For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. For <i>destination</i>, enter the number of the network or host to which the packet is being sent. For <i>destination-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the destination. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	<p>show running-config</p> <p>Example:</p>	Verifies your entries.

	Command or Action	Purpose
	Device# show running-config	
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Controlling Source Information that Your Switch Originates

You can control the multicast source information that originates with your Device:

- Sources you advertise (based on your sources)
- Receivers of source information (based on knowing the requestor)

For more information, see the [Redistributing Sources, on page 30](#) and the [Filtering Source-Active Request Messages, on page 32](#).

Redistributing Sources

SA messages originate on RPs to which sources have registered. By default, any source that registers with an RP is advertised. The *A flag* is set in the RP when a source is registered, which means the source is advertised in an SA unless it is filtered.

Follow these steps to further restrict which registered sources are advertised:

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 msdp redistribute [list access-list-name] [asn aspath-access-list-number] [route-map map] Example:	Configures which (S,G) entries from the multicast routing table are advertised in SA messages. By default, only sources within the local domain are advertised.

	Command or Action	Purpose
	<pre>Device(config)# ip msdp redistribute list 21</pre>	<ul style="list-style-type: none"> • (Optional) list <i>access-list-name</i>— Enters the name or number of an IP standard or extended access list. The range is 1 to 99 for standard access lists and 100 to 199 for extended lists. The access list controls which local sources are advertised and to which groups they send. • (Optional) asn <i>aspath-access-list-number</i>—Enters the IP standard or extended access list number in the range 1 to 199. This access list number must also be configured in the ip as-path access-list command. • (Optional) route-map <i>map</i>—Enters the IP standard or extended access list number in the range 1 to 199. This access list number must also be configured in the ip as-path access-list command. <p>The Device advertises (S,G) pairs according to the access list or autonomous system path access list.</p>
Step 4	<p>Use one of the following:</p> <ul style="list-style-type: none"> • access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] • access-list <i>access-list-number</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i> <p>Example:</p> <pre>Device(config)# access list 21 permit 194.1.22.0</pre> <p>or</p> <pre>Device(config)# access list 21 permit ip 194.1.22.0 10.1.1.10 194.3.44.0 10.1.1.10</pre>	<p>Creates an IP standard access list, repeating the command as many times as necessary.</p> <p>or</p> <p>Creates an IP extended access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> • access-list-number—Enters the same number created in Step 2. The range is 1 to 99 for standard access lists and 100 to 199 for extended lists. • deny—Denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. • protocol—Enters ip as the protocol name. • source—Enters the number of the network or host from which the packet is being sent. • source-wildcard—Enters the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • <i>destination</i>—Enters the number of the network or host to which the packet is being sent. • <i>destination-wildcard</i>—Enters the wildcard bits in dotted decimal notation to be applied to the destination. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Filtering Source-Active Request Messages

By default, only Devices that are caching SA information can respond to SA requests. By default, such a Device honors all SA request messages from its MSDP peers and supplies the IP addresses of the active sources.

However, you can configure the Device to ignore all SA requests from an MSDP peer. You can also honor only those SA request messages from a peer for groups described by a standard access list. If the groups in the access list pass, SA request messages are accepted. All other such messages from the peer for other groups are ignored.

To return to the default setting, use the **no ip msdp filter-sa-request** *{ip-address| name}* global configuration command.

Follow these steps to configure one of these options:

Procedure

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

	Command or Action	Purpose
	Example: Device> enable	<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	Use one of the following: <ul style="list-style-type: none"> • <code>ip msdp filter-sa-request {ip-addressname}</code> • <code>ip msdp filter-sa-request {ip-addressname} list access-list-number</code> Example: Device(config)# ip msdp filter sa-request 171.69.2.2	Filters all SA request messages from the specified MSDP peer. or Filters SA request messages from the specified MSDP peer for groups that pass the standard access list. The access list describes a multicast group address. The range for the access-list-number is 1 to 99.
Step 4	access-list access-list-number {deny permit} source [source-wildcard] Example: Device(config)# access-list 1 permit 192.4.22.0 0.0.0.255	Creates an IP standard access list, repeating the command as many times as necessary. <ul style="list-style-type: none"> • For <i>access-list-number</i>, the range is 1 to 99. • The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. • For <i>source</i>, enter the number of the network or host from which the packet is being sent. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Controlling Source Information that Your Switch Forwards

By default, the Device forwards all SA messages it receives to all its MSDP peers. However, you can prevent outgoing messages from being forwarded to a peer by using a filter or by setting a time-to-live (TTL) value.

Using a Filter

By creating a filter, you can perform one of these actions:

- Filter all source/group pairs
- Specify an IP extended access list to pass only certain source/group pairs
- Filter based on match criteria in a route map

Follow these steps to apply a filter:

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	Use one of the following: <ul style="list-style-type: none"> • ip msdp sa-filter out 	<ul style="list-style-type: none"> • Filters all SA messages to the specified MSDP peer.

	Command or Action	Purpose
	<pre>{ip-address name} • ip msdp sa-filter out {ip-address name} list access-list-number • ip msdp sa-filter out {ip-address name} route-map map-tag</pre> <p>Example:</p> <pre>Device(config)# ip msdp sa-filter out switch.cisco.com</pre> <p>or</p> <pre>Device(config)# ip msdp sa-filter out list 100</pre> <p>or</p> <pre>Device(config)# ip msdp sa-filter out switch.cisco.com route-map 22</pre>	<ul style="list-style-type: none"> • Passes only those SA messages that pass the IP extended access list to the specified peer. The range for the extended <i>access-list-number</i> is 100 to 199. <p>If both the list and the route-map keywords are used, all conditions must be true to pass any (S,G) pair in outgoing SA messages.</p> <ul style="list-style-type: none"> • Passes only those SA messages that meet the match criteria in the route map <i>map-tag</i> to the specified MSDP peer. <p>If all match criteria are true, a permit from the route map passes routes through the filter. A deny filters routes.</p>
Step 4	<p>access-list <i>access-list-number</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i></p> <p>Example:</p> <pre>Device(config)# access list 100 permit ip 194.1.22.0 10.1.1.10 194.3.44.0 10.1.1.10</pre>	<p>(Optional) Creates an IP extended access list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> • For <i>access-list-number</i>, enter the number specified in Step 2. • The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. • For <i>protocol</i>, enter ip as the protocol name. • For <i>source</i>, enter the number of the network or host from which the packet is being sent. • For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. • For <i>destination</i>, enter the number of the network or host to which the packet is being sent. • For <i>destination-wildcard</i>, enter the wildcard bits in dotted decimal notation to

	Command or Action	Purpose
		be applied to the destination. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Using TTL to Limit the Multicast Data Sent in SA Messages

You can use a TTL value to control what data is encapsulated in the first SA message for every source. Only multicast packets with an IP-header TTL greater than or equal to the *ttl* argument are sent to the specified MSDP peer. For example, you can limit internal traffic to a TTL of 8. If you want other groups to go to external locations, you must send those packets with a TTL greater than 8.

Follow these steps to establish a TTL threshold:

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: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ip msdp ttl-threshold <i>{ip-address name}</i> <i>ttl</i> Example: <pre>Device(config)# ip msdp ttl-threshold switch.cisco.com 0</pre>	Limits which multicast data is encapsulated in the first SA message to the specified MSDP peer. <ul style="list-style-type: none"> For <i>ip-address name</i>, enter the IP address or name of the MSDP peer to which the TTL limitation applies. For <i>ttl</i>, enter the TTL value. The default is 0, which means all multicast data packets are forwarded to the peer until the TTL is exhausted. The range is 0 to 255.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Controlling Source Information that Your Switch Receives

By default, the Device receives all SA messages that its MSDP RPF peers send to it. However, you can control the source information that you receive from MSDP peers by filtering incoming SA messages. In other words, you can configure the Device to not accept them.

You can perform one of these actions:

- Filter all incoming SA messages from an MSDP peer
- Specify an IP extended access list to pass certain source/group pairs
- Filter based on match criteria in a route map

Follow these steps to apply a filter:

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	Use one of the following: <ul style="list-style-type: none"> • ip msdp sa-filter in <pre>{ip-address name}</pre> • ip msdp sa-filter in <pre>{ip-address name} list access-list-number</pre> • ip msdp sa-filter in <pre>{ip-address name} route-map map-tag</pre> Example: <pre>Device(config)# ip msdp sa-filter in switch.cisco.com</pre> or <pre>Device(config)# ip msdp sa-filter in list 100</pre> or <pre>Device(config)# ip msdp sa-filter in switch.cisco.com route-map 22</pre>	<ul style="list-style-type: none"> • Filters all SA messages to the specified MSDP peer. • Passes only those SA messages from the specified peer that pass the IP extended access list. The range for the extended <i>access-list-number</i> is 100 to 199. If both the list and the route-map keywords are used, all conditions must be true to pass any (S,G) pair in outgoing SA messages. <ul style="list-style-type: none"> • Passes only those SA messages from the specified MSDP peer that meet the match criteria in the route map <i>map-tag</i>. If all match criteria are true, a permit from the route map passes routes through the filter. A deny filters routes.
Step 4	access-list access-list-number {deny permit} protocol source source-wildcard destination destination-wildcard Example: <pre>Device(config)# access list 100 permit ip 194.1.22.0 10.1.1.10 194.3.44.0 10.1.1.10</pre>	(Optional) Creates an IP extended access list, repeating the command as many times as necessary. <ul style="list-style-type: none"> • <i>access-list-number</i>, enter the number specified in Step 2. • The deny keyword denies access if the conditions are matched. The permit

	Command or Action	Purpose
		<p>keyword permits access if the conditions are matched.</p> <ul style="list-style-type: none"> • For <i>protocol</i>, enter ip as the protocol name. • For <i>source</i>, enter the number of the network or host from which the packet is being sent. • For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. • For <i>destination</i>, enter the number of the network or host to which the packet is being sent. • For <i>destination-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the destination. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring an MSDP Mesh Group

An MSDP mesh group is a group of MSDP speakers that have fully meshed MSDP connectivity among one another. Any SA messages received from a peer in a mesh group are not forwarded to other peers in the same mesh group. Thus, you reduce SA message flooding and simplify peer-RPF flooding. Use the **ip msdp**

mesh-group global configuration command when there are multiple RPs within a domain. It is especially used to send SA messages across a domain. You can configure multiple mesh groups (with different names) in a single Device.

Follow these steps to create a mesh group:

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp mesh-group name {ip-address name} Example: <pre>Device(config)# ip msdp mesh-group 2 switch.cisco.com</pre>	Configures an MSDP mesh group, and specifies the MSDP peer belonging to that mesh group. By default, the MSDP peers do not belong to a mesh group. <ul style="list-style-type: none"> • For <i>name</i>, enter the name of the mesh group. • For <i>ip-address name</i>, enter the IP address or name of the MSDP peer to be a member of the mesh group. Repeat this procedure on each MSDP peer in the group.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	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>	

Shutting Down an MSDP Peer

If you want to configure many MSDP commands for the same peer and you do not want the peer to become active, you can shut down the peer, configure it, and later bring it up. When a peer is shut down, the TCP connection is terminated and is not restarted. You can also shut down an MSDP session without losing configuration information for the peer.

Follow these steps to shut down a peer:

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	ip msdp shutdown {peer-name peer address} Example: Device(config)# <code>ip msdp shutdown switch.cisco.com</code>	Shuts down the specified MSDP peer without losing configuration information. For <i>peer-name peer address</i> , enter the IP address or name of the MSDP peer to shut down.
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.

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

Including a Bordering PIM Dense-Mode Region in MSDP

You can configure MSDP on a Device that borders a PIM sparse-mode region with a dense-mode region. By default, active sources in the dense-mode region do not participate in MSDP.



Note We do not recommend using the **ip msdp border sa-address** global configuration command. It is better to configure the border router in the sparse-mode domain to proxy-register sources in the dense-mode domain to the RP of the sparse-mode domain and have the sparse-mode domain use standard MSDP procedures to advertise these sources.

The **ip msdp originator-id** global configuration command also identifies an interface to be used as the RP address. If both the **ip msdp border sa-address** and the **ip msdp originator-id** global configuration commands are configured, the address derived from the **ip msdp originator-id** command specifies the RP address.

Follow these steps to configure the border router to send SA messages for sources active in the dense-mode region to the MSDP peers:

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp border sa-address interface-id Example: <pre>Device(config)# ip msdp border sa-address 0/1</pre>	Configures the switch on the border between a dense-mode and sparse-mode region to send SA messages about active sources in the dense-mode region. For <i>interface-id</i> , specifies the interface from which the IP address is derived and used as the RP address in SA messages.

	Command or Action	Purpose
		The IP address of the interface is used as the Originator-ID, which is the RP field in the SA message.
Step 4	ip msdp redistribute [<i>list access-list-name</i>] [<i>asn aspath-access-list-number</i>] [<i>route-map map</i>] Example: Device(config)# ip msdp redistribute list 100	Configures which (S,G) entries from the multicast routing table are advertised in SA messages. For more information, see the Redistributing Sources, on page 30 .
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring an Originating Address other than the RP Address

You can allow an MSDP speaker that originates an SA message to use the IP address of the interface as the RP address in the SA message by changing the Originator ID. You might change the Originator ID in one of these cases:

- If you configure a logical RP on multiple Device in an MSDP mesh group.
- If you have a Device that borders a PIM sparse-mode domain and a dense-mode domain. If a Device borders a dense-mode domain for a site, and sparse-mode is being used externally, you might want dense-mode sources to be known to the outside world. Because this Device is not an RP, it would not have an RP address to use in an SA message. Therefore, this command provides the RP address by specifying the address of the interface.

If both the **ip msdp border sa-address** and the **ip msdp originator-id** global configuration commands are configured, the address derived from the **ip msdp originator-id** command specifies the address of the RP.

Follow these steps to allow an MSDP speaker that originates an SA message to use the IP address on the interface as the RP address in the SA message:

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip msdp originator-id <i>interface-id</i> Example: <pre>Device(config)# ip msdp originator-id 0/1</pre>	Configures the RP address in SA messages to be the address of the originating device interface. For <i>interface-id</i> , specify the interface on the local Device.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining MSDP

Commands that monitor MSDP SA messages, peers, state, and peer status:

Table 2: Commands for Monitoring and Maintaining MSDP

Command	Purpose
debug ip msdp [<i>peer-address</i> <i>name</i>] [detail] [routes]	Debugs an MSDP activity.
debug ip msdp resets	Debugs MSDP peer reset reasons.
show ip msdp count [<i>autonomous-system-number</i>]	Displays the number of sources and groups originated in SA messages from each autonomous system. The ip msdp cache-sa-state command must be configured for this command to produce any output.
show ip msdp peer [<i>peer-address</i> <i>name</i>]	Displays detailed information about an MSDP peer.
show ip msdp sa-cache [<i>group-address</i> <i>source-address</i> <i>group-name</i> <i>source-name</i>] [<i>autonomous-system-number</i>]	Displays (S,G) state learned from MSDP peers.
show ip msdp summary	Displays MSDP peer status and SA message counts.

Commands that clear MSDP connections, statistics, and SA cache entries:

Table 3: Commands for Clearing MSDP Connections, Statistics, or SA Cache Entries

Command	Purpose
clear ip msdp peer <i>peer-address</i> <i>name</i>	Clears the TCP connection to the specified MSDP peer, resetting all MSDP message counters.
clear ip msdp statistics [<i>peer-address</i> <i>name</i>]	Clears statistics counters for one or all the MSDP peers without resetting the sessions.
clear ip msdp sa-cache [<i>group-address</i> <i>name</i>]	Clears the SA cache entries for all entries, all sources for a specific group, or all entries for a specific source/group pair.

Configuration Examples for Configuring MSDP

Configuring a Default MSDP Peer: Example

This example shows a partial configuration of Router A and Router C in . Each of these ISPs have more than one customer (like the customer in) who use default peering (no BGP or MBGP). In that case, they might have similar configurations. That is, they accept SAs only from a default peer if the SA is permitted by the corresponding prefix list.

Router A

```
Router(config)# ip msdp default-peer 10.1.1.1
Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a
```

```
Router(config)# ip prefix-list site-b permit 10.0.0.0/1
```

Router C

```
Router(config)# ip msdp default-peer 10.1.1.1 prefix-list site-a
Router(config)# ip prefix-list site-b permit 10.0.0.0/1
```

Caching Source-Active State: Example

This example shows how to enable the cache state for all sources in 171.69.0.0/16 sending to groups 224.2.0.0/16:

```
Device(config)# ip msdp cache-sa-state 100
Device(config)# access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255
```

Controlling Source Information that Your Switch Originates: Example

This example shows how to configure the switch to filter SA request messages from the MSDP peer at 171.69.2.2. SA request messages from sources on network 192.4.22.0 pass access list 1 and are accepted; all others are ignored.

```
Device(config)# ip msdp filter sa-request 171.69.2.2 list 1
Device(config)# access-list 1 permit 192.4.22.0 0.0.0.255
```

Controlling Source Information that Your Switch Forwards: Example

This example shows how to allow only (S,G) pairs that pass access list 100 to be forwarded in an SA message to the peer named *switch.cisco.com*:

```
Device(config)# ip msdp peer switch.cisco.com connect-source gigabitethernet1/0/1
Device(config)# ip msdp sa-filter out switch.cisco.com list 100
Device(config)# access-list 100 permit ip 171.69.0.0 0.0.255.255 224.2.0.0 0.0.255.255
```

Controlling Source Information that Your Switch Receives: Example

This example shows how to filter all SA messages from the peer named *switch.cisco.com*:

```
Device(config)# ip msdp peer switch.cisco.com connect-source gigabitethernet1/0/1
Device(config)# ip msdp sa-filter in switch.cisco.com
```

Feature Information for Multicast Source Discovery Protocol

Table 4: Feature Information for Multicast Source Discovery Protocol

Feature Name	Release	Feature Information
Multicast Source Discovery Protocol	Cisco IOS XE Everest 16.5.1a	This feature was introduced



CHAPTER 3

Configuring IP Unicast Routing

- [Restrictions for IP Unicast Routing, on page 49](#)
- [Information About Configuring IP Unicast Routing, on page 49](#)
- [Information About IP Routing, on page 50](#)
- [How to Configure IP Routing, on page 56](#)
- [How to Configure IP Addressing, on page 57](#)
- [Monitoring and Maintaining IP Addressing, on page 75](#)
- [How to Configure IP Unicast Routing, on page 76](#)
- [Monitoring and Maintaining the IP Network, on page 77](#)
- [Feature Information for IP Unicast Routing, on page 77](#)

Restrictions for IP Unicast Routing

- On enabling IP routing, the VLAN configured as SVI will also learn broadcast ARP requests which are not self destined.
- The number of routed ports and SVIs that you can configure is 4000. Exceeding the recommended number and volume of features being implemented might impact CPU utilization because of hardware limitations.
- 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 and the Routing Information Protocol (RIP), are available with both the Network Essentials license and the Network Advantage license. To use advanced routing features and other routing protocols, you must have the Network Advantage license enabled on the standalone switch or on the active switch.



Note

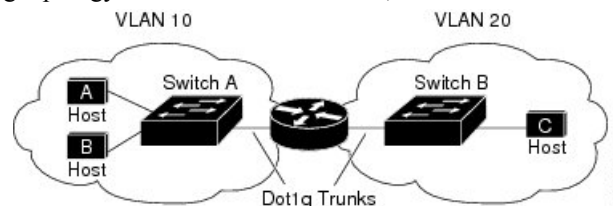
In addition to IPv4 traffic, you can also enable IP Version 6 (IPv6) unicast routing and configure interfaces to forward IPv6 traffic if the switch or switch stack is running the Network Essentials or Network Advantage license.

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 2: 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.

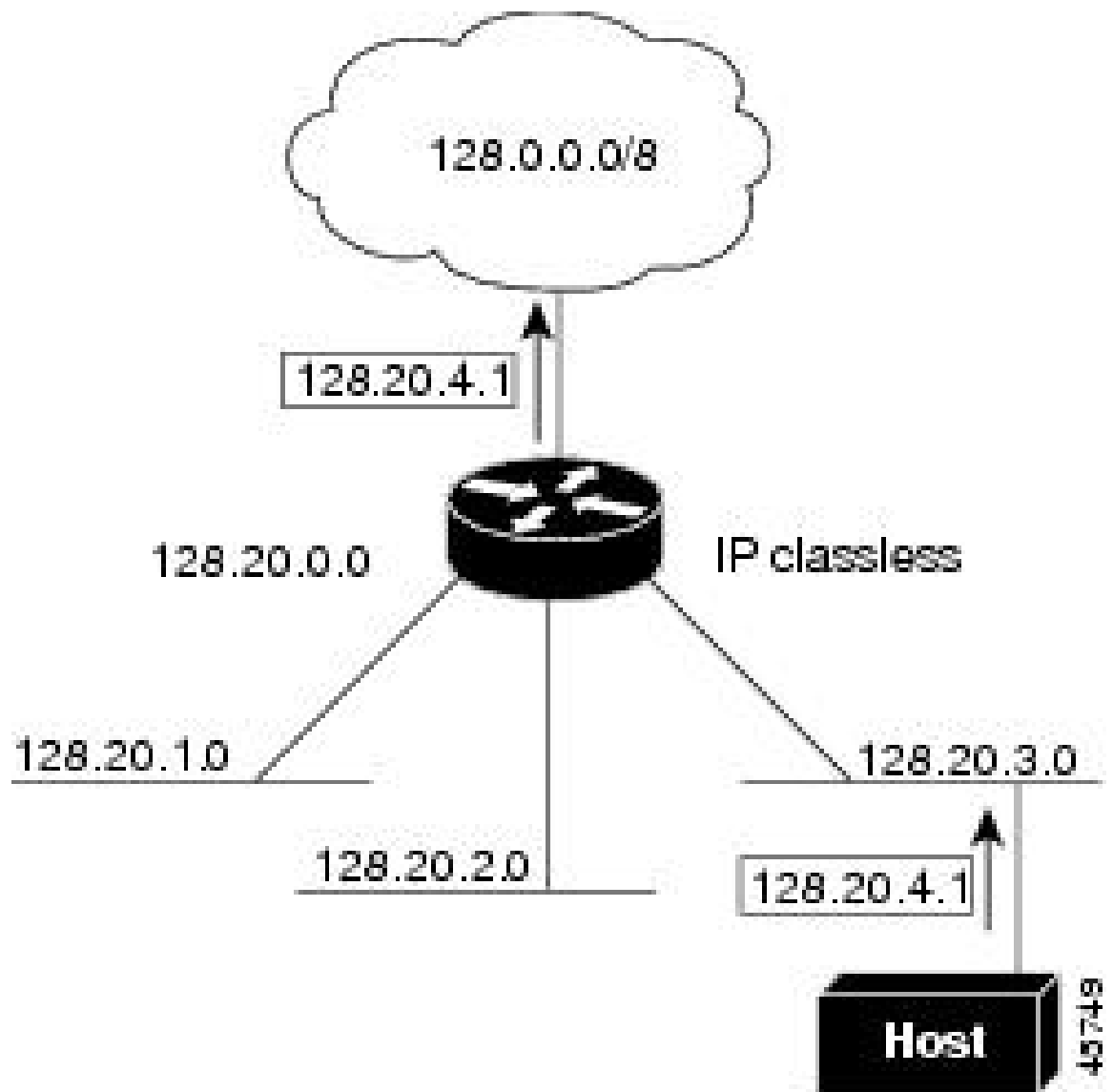
- Routers using link-state protocols maintain a complex database of network topology, based on the exchange of link-state advertisements (LSAs) between routers. LSAs are triggered by an event in the network, which speeds up the convergence time or time required to respond to these changes. Link-state 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 and Border Gateway Protocol (BGP), which adds a path vector mechanism. 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

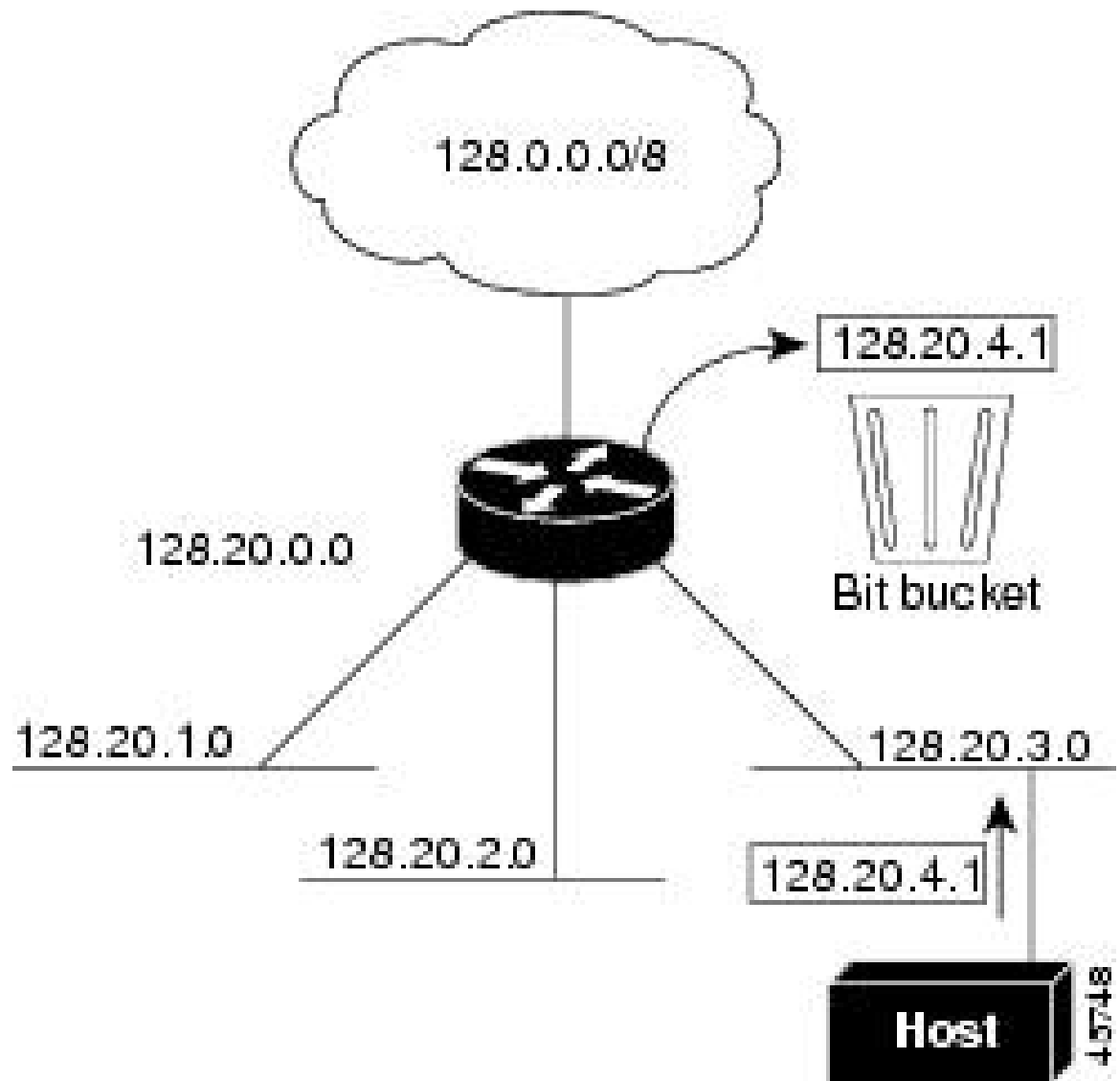
By default, classless routing behavior is enabled on the Device when it is configured to route. With classless routing, if a router receives packets for a subnet of a network with no default route, the router forwards the packet to the best supernet route. A supernet consists of contiguous blocks of 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 3: 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 4: 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 5: 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, UI 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: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	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: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	no ip classless Example: <pre>Device(config)#no ip classless</pre>	Disables classless routing behavior.
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show running-config Example: <pre>Device# show running-config</pre>	Verifies your entries.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	arp ip-address hardware-address type Example: <pre>Device(config)# ip 10.1.5.1 c2f3.220a.12f4 arpa</pre>	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: <pre>Device(config)# ip 10.1.5.3 d7f3.220d.12f5 arpa alias</pre>	(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: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the interface to configure.
Step 6	arp timeout seconds Example: <pre>Device(config-if)# arp timeout 20000</pre>	(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

	Command or Action	Purpose
		timeout. Be aware that decreasing the ARP timeout can result in increased network traffic. Setting the ARP timeout to 60 seconds or less is generally not recommended as it may cause network disruptions.
Step 7	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 8	show interfaces [<i>interface-id</i>] Example: <pre>Device# show interfaces gigabitethernet 1/0/1</pre>	Verifies the type of ARP and the timeout value used on all interfaces or a specific interface.
Step 9	show arp Example: <pre>Device# show arp</pre>	Views the contents of the ARP cache.
Step 10	show ip arp Example: <pre>Device# show ip arp</pre>	Views the contents of the ARP cache.
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <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	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/2	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	arp arpa Example: Device(config-if)# arp arpa	Specifies the ARP encapsulation method:
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show interfaces [<i>interface-id</i>] Example: Device# show interfaces	Verifies ARP encapsulation configuration on all interfaces or the specified interface.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(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	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 <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/2</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip proxy-arp Example: <pre>Device(config-if)# ip proxy-arp</pre>	Enables proxy ARP on the interface.
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show ip interface [<i>interface-id</i>] Example: <pre>Device# show ip interface gigabitethernet 1/0/2</pre>	Verifies the configuration on the interface or all interfaces.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	ip default-gateway ip-address Example: <pre>Device(config)# ip default gateway 10.1.5.1</pre>	Sets up a default gateway (router).
Step 4	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip redirects Example: <pre>Device# show ip redirects</pre>	Displays the address of the default gateway router to verify the setting.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config</pre>	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	<code>startup-config</code>	

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 Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip irdp Example: <pre>Device(config-if)# ip irdp</pre>	Enables IRDP processing on the interface.
Step 5	ip irdp multicast Example: <pre>Device(config-if)# ip irdp multicast</pre>	(Optional) Sends IRDP advertisements to the multicast address (224.0.0.1) instead of IP broadcasts.

	Command or Action	Purpose
		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.
Step 6	ip irdp holdtime <i>seconds</i> Example: Device(config-if)# ip irdp holdtime 1000	(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	ip irdp maxadvertinterval <i>seconds</i> Example: Device(config-if)# ip irdp maxadvertinterval 650	(Optional) Sets the IRDP maximum interval between advertisements. The default is 600 seconds.
Step 8	ip irdp minadvertinterval <i>seconds</i> Example: Device(config-if)# ip irdp minadvertinterval 500	(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).
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 [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.

	Command or Action	Purpose
Step 12	show ip irdp Example: <pre>Device# show ip irdp</pre>	Verifies settings by displaying IRDP values.
Step 13	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <pre>Device> enable</pre>	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# configure terminal	
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 UDP 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.
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.

	Command or Action	Purpose
Step 10	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip helper-address <i>address</i> Example: <pre>Device(config-if)# ip helper address 10.1.10.1</pre>	Enables forwarding and specifies the destination address for forwarding UDP broadcast packets, including BOOTP.
Step 5	exit Example: <pre>Device(config-if)# exit</pre>	Returns to global configuration mode.
Step 6	ip forward-protocol {udp [<i>port</i>] nd sdns} Example:	Specifies which protocols the router forwards when forwarding broadcast packets.

	Command or Action	Purpose
	Device(config)# ip forward-protocol sdns	
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. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

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

Flooding IP Broadcasts

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.

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

	Command or Action	Purpose
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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 6: 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.
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 7: 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 interfaces using 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 using RIP as the routing protocol :

```
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 8: Commands to Clear IP Routes or Display Route Status

Command	Purpose
show ip route summary	Displays the current state of the routing table in summary

Feature Information for IP Unicast Routing

Table 9: Feature Information for IP Unicast Routing

Feature Name	Release	Feature Information
IP Unicast Routing	Cisco IOS XE Everest 16.5.1a	This feature was introduced



CHAPTER 4

Protocol-Independent Features

- [Protocol-Independent Features, on page 79](#)

Protocol-Independent Features

This section describes IP routing protocol-independent features that are available on switches running the Network Essentials 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# configure terminal	Enters global configuration mode.
Step 2	ip cef Example: Device(config)# ip cef	Enables CEF operation on a non-stacking switch. Go to Step 4.
Step 3	ip cef distributed Example: Device(config)# ip cef distributed	Enables CEF operation on a active switch.
Step 4	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 5	ip route-cache cef Example: Device(config-if)# ip route-cache cef	Enables CEF on the interface for software-forwarded traffic.

	Command or Action	Purpose
Step 6	end Example: <pre>Device(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 7	show ip cef Example: <pre>Device# show ip cef</pre>	Displays the CEF status on all interfaces.
Step 8	show cef linecard [detail] Example: <pre>Device# show cef linecard detail</pre>	(Optional) Displays CEF-related interface information on a non-stacking switch.
Step 9	show cef linecard [slot-number] [detail] Example: <pre>Device# show cef linecard 5 detail</pre>	(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: <pre>Device# show cef interface gigabitethernet 1/0/1</pre>	Displays detailed CEF information for all interfaces or the specified interface.
Step 11	show adjacency Example: <pre>Device# show adjacency</pre>	Displays CEF adjacency table information.
Step 12	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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:

Procedure

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

	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-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:

Procedure

	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 [<i>id</i>] } Example:	Selects a CEF load-balancing algorithm. <ul style="list-style-type: none"> The original keyword sets the load-balancing algorithm to the original

	Command or Action	Purpose
	<pre>Device(config)# ip cef load-sharing algorithm universal</pre>	<p>algorithm, based on a source IP and destination IP hash.</p> <ul style="list-style-type: none"> 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	<pre>end</pre> <p>Example:</p> <pre>Device(config)# end</pre>	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 maximum 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: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

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 10: 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
Internal BGP	200
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:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	ip route prefix mask {address interface} [distance] Example: Device(config)# <code>ip route prefix mask gigabitethernet 1/0/4</code>	Establish a static route.
Step 4	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 5	show ip route Example: Device# <code>show ip route</code>	Displays the current state of the routing table to verify the configuration.
Step 6	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(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# configure terminal	Enters global configuration mode.
Step 2	ip default-network <i>network number</i> Example: Device(config)# ip default-network 1	Specifies a default network.
Step 3	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 4	show ip route Example: Device# show ip route	Displays the selected default route in the gateway of last resort display.
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:	Defines any route maps used to control redistribution and enter route-map configuration mode.

	Command or Action	Purpose
	<pre>Device(config)# route-map rip-to-ospf permit 4</pre>	<p><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.</p> <p>(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.</p> <p><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.</p>
Step 3	<p>match as-path <i>path-list-number</i></p> <p>Example:</p> <pre>Device(config-route-map)#match as-path 10</pre>	Matches a BGP AS path access list.
Step 4	<p>match community-list <i>community-list-number</i> [exact]</p> <p>Example:</p> <pre>Device(config-route-map)# match community-list 150</pre>	Matches a BGP community list.
Step 5	<p>match ip address {<i>access-list-number</i> <i>access-list-name</i>} [...<i>access-list-number</i> ...<i>access-list-name</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address 5 80</pre>	Matches a standard access list by specifying the name or number. It can be an integer from 1 to 199.
Step 6	<p>match metric <i>metric-value</i></p> <p>Example:</p> <pre>Device(config-route-map)# match metric 2000</pre>	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	<p>match ip next-hop {<i>access-list-number</i> <i>access-list-name</i>} [...<i>access-list-number</i> ...<i>access-list-name</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip next-hop 8 45</pre>	Matches a next-hop router address passed by one of the access lists specified (numbered from 1 to 199).

	Command or Action	Purpose
Step 8	match tag <i>tag value</i> [... <i>tag-value</i>] Example: <pre>Device(config-route-map)# match tag 3500</pre>	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 interface <i>type number</i> [... <i>type-number</i>] Example: <pre>Device(config-route-map)# match interface gigabitethernet 1/0/1</pre>	Matches the specified next hop route out one of the specified interfaces.
Step 10	match ip route-source { <i>access-list-number</i> <i>access-list-name</i> } [... <i>access-list-number</i> ... <i>access-list-name</i>] Example: <pre>Device(config-route-map)# match ip route-source 10 30</pre>	Matches the address specified by the specified advertised access lists.
Step 11	match route-type { local internal external [type-1 type-2]} Example: <pre>Device(config-route-map)# match route-type local</pre>	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 <i>halflife reuse suppress max-suppress-time</i> Example: <pre>Device(config-route-map)# set dampening 30 1500 10000 120</pre>	Sets BGP route dampening factors.
Step 13	set local-preference <i>value</i> Example: <pre>Device(config-route-map)# set local-preference 100</pre>	Assigns a value to a local BGP path.
Step 14	set origin { igp egp <i>as</i> incomplete } Example: <pre>Device(config-route-map)# set origin igp</pre>	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.

	Command or Action	Purpose
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.
Step 21	set weight <i>number</i> Example: <pre>Device(config-route-map)# set weight 100</pre>	Sets the BGP weight for the routing table. The value can be from 1 to 65535.
Step 22	end Example: <pre>Device(config-route-map)# end</pre>	Returns to privileged EXEC mode.
Step 23	show route-map Example: <pre>Device# show route-map</pre>	Displays all route maps configured or only the one specified to verify configuration.
Step 24	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <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	redistribute protocol [process-id] {level-1 level-1-2 level-2} [metric metric-value] [metric-type type-value] [match internal external type-value] [tag tag-value] [route-map map-tag] [weight weight] [subnets] Example: <pre>Device(config-router)# redistribute eigrp 1</pre>	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 number Example: <pre>Device(config-router)# default-metric 1024</pre>	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: <pre>Device(config-router)# default-metric 1000 100 250 100 1500</pre>	Cause the EIGRP routing protocol to use the same metric value for all non-EIGRP redistributed routes.
Step 6	end Example: <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 7	show route-map Example: <pre>Device# show route-map</pre>	Displays all route maps configured or only the one specified to verify configuration.
Step 8	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

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

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

- To use PBR, you must have the Network Essentials license enabled on the switch or active switch.
- 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, or local packets, are not normally policy-routed. When you globally enable local PBR on the switch, all packets that originate on the switch are subject to local PBR. Local PBR is disabled by default.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	route-map <i>map-tag</i> [permit] [<i>sequence number</i>] Example: <pre>Device(config)# route-map pbr-map permit</pre>	<p>Defines route maps that are used to control where packets are output, and enters route-map configuration mode.</p> <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. <p>Note Do not configure the route-map <i>map-tag</i> [<i>sequence number</i>] command without a match or set action defined for the sequence because this will result in incorrect traffic forwarding in the hardware.</p>
Step 3	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [<i>access-list-number</i> ... <i>access-list-name</i>] Example: <pre>Device(config-route-map)# match ip address 110 140</pre>	<p>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.</p> <p>If you do not specify a match command, the route map is applicable to all packets.</p>

	Command or Action	Purpose
Step 4	match length min max Example: Device(config-route-map)# match length 64 1500	Matches the length of the packet.
Step 5	set ip next-hop ip-address [...ip-address] 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 6	exit Example: Device(config-route-map)# exit	Returns to global configuration mode.
Step 7	interface interface-id Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to be configured.
Step 8	ip policy route-map map-tag 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 9	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 10	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 11	ip local policy route-map map-tag 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 12	end Example: Device(config)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	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 14	show ip policy Example: Device# show ip policy	(Optional) Displays policy route maps attached to the interface.
Step 15	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:	Enters router configuration mode.

	Command or Action	Purpose
	Device(config)# router ospf	
Step 3	passive-interface <i>interface-id</i> Example: <pre>Device(config-router)# passive-interface gigabitethernet 1/0/1</pre>	Suppresses sending routing updates through the specified Layer 3 interface.
Step 4	passive-interface default Example: <pre>Device(config-router)# passive-interface default</pre>	(Optional) Sets all interfaces as passive by default.
Step 5	no passive-interface <i>interface type</i> Example: <pre>Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5</pre>	(Optional) Activates only those interfaces that need to have adjacencies sent.
Step 6	network <i>network-address</i> Example: <pre>Device(config-router)# network 10.1.1.1</pre>	(Optional) Specifies the list of networks for the routing process. The <i>network-address</i> is an IP address.
Step 7	end Example: <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 8	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 2	router { rip eigrp } Example: <pre>Device(config)# router eigrp 10</pre>	Enters router configuration mode.
Step 3	distribute-list {access-list-number access-list-name} out [interface-name routing process autonomous-system-number] Example: <pre>Device(config-router)# distribute 120 out gigabitethernet 1/0/7</pre>	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: <pre>Device(config-router)# distribute-list 125 in</pre>	Suppresses processing in routes listed in updates.
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>	<p>Defines an administrative distance.</p> <p><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.</p> <p>(Optional) <i>ip access list</i>—An IP standard or extended access list to be applied to incoming routing updates.</p>
Step 4	end Example: <pre>Device(config-router)# end</pre>	Returns to privileged EXEC mode.
Step 5	show ip protocols Example: <pre>Device# show ip protocols</pre>	Displays the default administrative distance for a specified routing process.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(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.
Step 3	key <i>number</i> Example: Device(config-keychain)# key 2000	Identifies the key number. The range is 0 to 2147483647.
Step 4	key-string <i>text</i> Example: Device(config-keychain)# Room 20, 10th floor	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	accept-lifetime <i>start-time</i> {infinite <i>end-time</i> <i>duration seconds</i>} Example: Device(config-keychain)# accept-lifetime 12:30:00 Jan 25 1009 infinite	(Optional) Specifies the time period during which the key can be received. 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 .

	Command or Action	Purpose
Step 6	send-lifetime <i>start-time</i> { infinite <i>end-time</i> duration <i>seconds</i> } Example: <pre>Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite</pre>	(Optional) Specifies the time period during which the key can be sent. 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 .
Step 7	end Example: <pre>Device(config-keychain)# end</pre>	Returns to privileged EXEC mode.
Step 8	show key chain Example: <pre>Device# show key chain</pre>	Displays authentication key information.
Step 9	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.



CHAPTER 5

Implementing Multiprotocol BGP for IPv6

This module describes how to configure multiprotocol Border Gateway Protocol (BGP) for IPv6. BGP is an Exterior Gateway Protocol (EGP) used mainly to connect separate routing domains that contain independent routing policies (autonomous systems). Connecting to a service provider for access to the Internet is a common use for BGP. BGP can also be used within an autonomous system and this variation is referred to as internal BGP (iBGP). Multiprotocol BGP is an enhanced BGP that carries routing information for multiple network layer protocol address families, for example, IPv6 address family and for IP multicast routes. All BGP commands and routing policy capabilities can be used with multiprotocol BGP.

- [Information About Implementing Multiprotocol BGP for IPv6, on page 105](#)
- [How to Implement Multiprotocol BGP for IPv6, on page 106](#)
- [Verifying the IPv6 Multiprotocol BGP Configuration , on page 125](#)
- [Configuration Examples for Implementing Multiprotocol BGP for IPv6, on page 126](#)
- [Additional References for Implementing Multiprotocol BGP for IPv6, on page 129](#)
- [Feature Information for Implementing Multiprotocol BGP for IPv6, on page 129](#)

Information About Implementing Multiprotocol BGP for IPv6

Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP is the supported Exterior Gateway Protocol (EGP) for IPv6. Multiprotocol BGP extensions for IPv6 supports many of the same features and functionality as IPv4 BGP. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and Network Layer Reachability Information (NLRI) and next hop (the next device in the path to the destination) attributes that use IPv6 addresses.

IPv6 Multiprotocol BGP Peering Using a Link-Local Address

The IPv6 multiprotocol BGP can be configured between two IPv6 devices (peers) using link-local addresses. For this function to work, you must identify the interface for the neighbor by using the **neighbor update-source** command, and you must configure a route map to set an IPv6 global next hop.

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP

include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are usable only for IP unicast, not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family

The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco nonstop forwarding (NSF) functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state.

NSF continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. Forwarding is maintained by synchronizing the FIB between the active and standby RP. On switchover, forwarding is maintained using the FIB. The RIB is not kept synchronized; therefore, the RIB is empty on switchover. The RIB is repopulated by the routing protocols and subsequently informs FIB about RIB convergence by using the NSF_RIB_CONVERGED registry call. The FIB tables are updated from the RIB, removing any stale entries. The RIB starts a failsafe timer during RP switchover, in case the routing protocols fail to notify the RIB of convergence.

The Cisco BGP address family identifier (AFI) model is designed to be modular and scalable, and to support multiple AFI and subsequent address family identifier (SAFI) configurations.

How to Implement Multiprotocol BGP for IPv6

Configuring an IPv6 BGP Routing Process and BGP Router ID

Perform this task to configure an IPv6 BGP routing process and an optional BGP router ID for a BGP-speaking device.

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is 32-bit value that is often represented by an IPv4 address. By default, the router ID is set to the IPv4 address of a loopback interface on the device. If no loopback interface is configured on the device, then the software chooses the highest IPv4 address configured to a physical interface on the device to represent the BGP router ID.

When configuring BGP on a device that is enabled only for IPv6 (that is, the device does not have an IPv4 address), you must manually configure the BGP router ID for the device. The BGP router ID, which is represented as a 32-bit value using an IPv4 address syntax, must be unique to the BGP peers of the device.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Configures a BGP routing process, and enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: <pre>Device(config-router)# no bgp default ipv4-unicast</pre>	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as command unless you configure the no bgp default ipv4-unicast command before configuring the neighbor remote-as command.
Step 5	bgp router-id <i>ip-address</i> Example: <pre>Device(config-router)# bgp router-id 192.168.99.70</pre>	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.
Step 6	end Example: <pre>Device(config-router)# end</pre>	Exits router configuration mode and returns to privileged EXEC mode.

Configuring IPv6 Multiprotocol BGP Between Two Peers

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor {<i>ip-address</i> <i>ipv6-address</i> [%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [<i>alternate-as autonomous-system-number</i> ...] Example: <pre>Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600</pre>	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	address-family ipv6 [<i>unicast</i> <i>multicast</i>] Example: <pre>Device(config-router)# address-family ipv6</pre>	Specifies the IPv6 address family and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> %} activate Example:	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.

	Command or Action	Purpose
	Device(config-router-af) # neighbor 2001:DB8:0:CC00::1 activate	
Step 7	end Example: Device(config-router-af) # end	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring IPv6 Multiprotocol BGP Between Two Peers Using Link-Local Addresses

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

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	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i> Example: Device(config-router)# neighbor	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose
	<pre>2001:DB8:0000:0000:0000:0000:0111 remote-as 64600</pre>	<ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be a link-local IPv6 address in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 5	<p>neighbor {<i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i>} update-source <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0111 update-source gigabitethernet 0/0/0</pre>	<p>Specifies the link-local address over which the peering is to occur.</p> <ul style="list-style-type: none"> If there are multiple connections to the neighbor and you do not specify the neighbor interface by using the <i>interface-type</i> and <i>interface-number</i> arguments in the neighbor update-source command, a TCP connection cannot be established with the neighbor using link-local addresses.
Step 6	<p>address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn]</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv6</pre>	<p>Specifies the IPv6 address family, and enters address family configuration mode.</p> <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i>} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0111 activate</pre>	<p>Enables the neighbor to exchange prefixes for the IPv6 address family with the local router using the specified link-local addresses.</p>
Step 8	<p>neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i>} route-map <i>map-name</i> [in out]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0111 route-map nh6 out</pre>	<p>Applies a route map to incoming or outgoing routes.</p>

	Command or Action	Purpose
Step 9	exit Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode, and returns to router configuration mode.
Step 10	exit Example: <pre>Device(config-router)# exit</pre>	Exits router configuration mode, and returns to global configuration mode.
Step 11	route-map map-tag [permit deny] <i>[sequence-number]</i> Example: <pre>Device(config)# route-map nh6 permit 10</pre>	Defines a route map and enters route-map configuration mode.
Step 12	match ipv6 address {prefix-list <i>prefix-list-name access-list-name</i> Example: <pre>Device(config-route-map)# match ipv6 address prefix-list list1</pre>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.
Step 13	set ipv6 next-hop ipv6-address <i>[link-local-address] [peer-address]</i> Example: <pre>Device(config-route-map)# set ipv6 next-hop 2001:DB8::1</pre>	<p>Overrides the next hop advertised to the peer for IPv6 packets that pass a match clause of a route map for policy routing.</p> <ul style="list-style-type: none"> • The <i>ipv6-address</i> argument specifies the IPv6 global address of the next hop. It need not be an adjacent router. • The <i>link-local-address</i> argument specifies the IPv6 link-local address of the next hop. It must be an adjacent router.

	Command or Action	Purpose
		Note The route map sets the IPv6 next-hop addresses (global and link-local) in BGP updates. If the route map is not configured, the next-hop address in the BGP updates defaults to the unspecified IPv6 address (::), which is rejected by the peer. If you specify only the global IPv6 next-hop address (the <i>ipv6-address</i> argument) with the set ipv6 next-hop command after specifying the neighbor interface (the <i>interface-type</i> argument) with the neighbor update-source command in Step 5, the link-local address of the interface specified with the <i>interface-type</i> argument is included as the next-hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses.
Step 14	end Example: Device(config-route-map) # end	Exits route-map configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If peering is not established by this task, it may be because of a missing route map **set ipv6 next-hop** command. Use the **debug bgp ipv6 update** command to display debugging information on the updates to help determine the state of the peering.

Configuring an IPv6 Multiprotocol BGP Peer Group

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, peer groups that are defined in router configuration mode using the **neighbor peer-group** command exchange only IPv4 unicast address prefixes. To exchange other address prefix

types, such as IPv6 prefixes, you must activate peer groups using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

- Members of a peer group automatically inherit the address prefix configuration of the peer group.
- IPv4 active neighbors cannot exist in the same peer group as active IPv6 neighbors. Create separate peer groups for IPv4 peers and IPv6 peers.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified BGP routing process.
Step 4	neighbor <i>peer-group-name</i> peer-group Example: <pre>Device(config-router)# neighbor group1 peer-group</pre>	Creates a multiprotocol BGP peer group.
Step 5	neighbor {<i>ip-address</i> <i>ipv6-address</i>[%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [<i>alternate-as autonomous-system-number</i> ...] Example: <pre>Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600</pre>	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 6	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i> <i>vpn</i>] Example: <pre>Device(config-router)# address-family ipv6 unicast</pre>	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address %</i> } activate Example: <pre>Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate</pre>	<p>Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router.</p> <ul style="list-style-type: none"> To avoid extra configuration steps for each neighbor, use the neighbor activate command with the <i>peer-group-name</i> argument as an alternative in this step.
Step 8	neighbor <i>ip-address</i> <i>ipv6-address</i> } send-label Example: <pre>Device(config-router-af)# neighbor 192.168.99.70 send-label</pre>	<p>Advertises the capability of the device to send MPLS labels with BGP routes.</p> <ul style="list-style-type: none"> In IPv6 address family configuration mode, this command enables binding and advertisement of aggregate labels when advertising IPv6 prefixes in BGP.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> } peer-group <i>peer-group-name</i> Example: <pre>Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 peer-group group1</pre>	<p>Assigns the IPv6 address of a BGP neighbor to a peer group.</p>
Step 10	end Example: <pre>Device(config-router-af)# end</pre>	<p>Exits address family configuration mode, and returns to privileged EXEC mode.</p>

Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor {<i>ip-address</i> <i>ipv6-address</i>[%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [<i>alternate-as autonomous-system-number ...</i>] Example: <pre>Device(config-router)# neighbor 2001:DB8:0:cc00::1 remote-as 64600</pre>	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i> <i>vpnv6</i>] Example: <pre>Device(config-router)# address-family ipv6</pre>	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} activate Example: <pre>Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 activate</pre>	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device using the specified link-local addresses.
Step 7	neighbor {<i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> [%]} route-map <i>map-name</i> {in out}	Applies a route map to incoming or outgoing routes.

	Command or Action	Purpose
	Example: <pre>Device(config-router-af)# neighbor 2001:DB8:0:cc00::1 route-map rtp in</pre>	<ul style="list-style-type: none"> Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 8	exit Example: <pre>Device(config-router-af)# exit</pre>	Exits address family configuration mode, and returns to router configuration mode.
Step 9	exit Example: <pre>Device(config-router)# exit</pre>	Exits router configuration mode, and returns to global configuration mode.
Step 10	route-map map-tag [permit deny] [sequence-number] Example: <pre>Device(config)# route-map rtp permit 10</pre>	Defines a route map and enters route-map configuration mode. <ul style="list-style-type: none"> Follow this step with a match command.
Step 11	match ipv6 address {prefix-list prefix-list-name access-list-name} Example: <pre>Device(config-route-map)# match ipv6 address prefix-list list1</pre>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.
Step 12	end Example: <pre>Device(config-route-map)# end</pre>	Exits route-map configuration mode and returns to privileged EXEC mode.

Redistributing Prefixes into IPv6 Multiprotocol BGP

Redistribution is the process of redistributing, or injecting, prefixes from one routing protocol into another routing protocol. This task explains how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

Procedure

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

	Command or Action	Purpose
	Example: Device> enable	<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 bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i> <i>vpn</i>] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	redistribute bgp [<i>process-id</i>] [<i>metric metric-value</i>] [<i>route-map map-name</i>] Example: Device(config-router-af)# redistribute bgp 64500 metric 5	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 6	end Example: Device(config-router-af)# end	Exits address family configuration mode, and returns to privileged EXEC mode.

Advertising Routes into IPv6 Multiprotocol BGP

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i> <i>vpn6</i>] Example: <pre>Device(config-router)# address-family ipv6 unicast</pre>	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	network {<i>network-number</i> [<i>mask network-mask</i>] <i>nsap-prefix</i>} [<i>route-map map-tag</i>] Example: <pre>Device(config-router-af)# network 2001:DB8::/24</pre>	Advertises (injects) the specified prefix into the IPv6 BGP database (the routes must first be found in the IPv6 unicast routing table). <ul style="list-style-type: none"> • The prefix is injected into the database for the address family specified in the previous step. • Routes are tagged from the specified prefix as “local origin.” • The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion

	Command or Action	Purpose
		of the address). A slash mark must precede the decimal value.
Step 6	exit Example: <pre>Device(config-router-af) # exit</pre>	Exits address family configuration mode, and returns the device to router configuration mode. <ul style="list-style-type: none"> • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Advertising IPv4 Routes Between IPv6 BGP Peers

If an IPv6 network is connecting two separate IPv4 networks, IPv6 can be used to advertise the IPv4 routes. Configure the peering using the IPv6 addresses within the IPv4 address family. Set the next hop with a static route or with an inbound route map because the advertised next hop will usually be unreachable. Advertising IPv6 routes between two IPv4 peers is also possible using the same model.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	neighbor <i>peer-group-name</i> peer-group Example: <pre>Device(config-router)# neighbor 6peers peer-group</pre>	Creates a multiprotocol BGP peer group.
Step 5	neighbor {<i>ip-address</i> <i>ipv6-address</i>[%] <i>peer-group-name</i>} remote-as <i>autonomous-system-number</i> [<i>alternate-as</i> <i>autonomous-system-number</i> ...] Example:	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.

	Command or Action	Purpose
	Device(config-router)# neighbor 6peers remote-as 65002	
Step 6	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Device(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor ipv6-address peer-group <i>peer-group-name</i> Example: Device(config-router-af)# neighbor 2001:DB8:1234::2 peer-group 6peers	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address [%]</i> } route-map <i>map-name</i> { in out } Example: Device(config-router-af)# neighbor 6peers route-map rmap out	Applies a route map to incoming or outgoing routes. • Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 9	exit Example: Device(config-router-af)# exit	Exits address family configuration mode, and returns the device to router configuration mode.
Step 10	exit Example: Device(config-router)# exit	Exits router configuration mode, and returns the device to global configuration mode.
Step 11	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# route-map rmap permit 10	Defines a route map and enters route-map configuration mode.
Step 12	set ip next-hop <i>ip-address</i> [... <i>ip-address</i>] [<i>peer-address</i>] Example:	Overrides the next hop advertised to the peer for IPv4 packets.

	Command or Action	Purpose
	Device(config-route-map)# set ip next-hop 10.21.8.10	
Step 13	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.

Assigning BGP Administrative Distance for Multicast BGP Routes

Perform this task to specify an administrative distance for multicast BGP routes to be used in RPF lookups for comparison with unicast routes.



Caution

Changing the administrative distance of BGP internal routes is not recommended. One problem that can occur is the accumulation of routing table inconsistencies, which can break routing.

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: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Device(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn] Example: Device(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. • The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.

	Command or Action	Purpose
		<ul style="list-style-type: none"> The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	distance bgp <i>external-distance</i> <i>internal-distance local-distance</i> Example: <pre>Device(config-router-af)# distance bgp 10 50 100</pre>	Configures the administrative distance for BGP routes.
Step 6	end Example: <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Generating IPv6 Multicast BGP Updates

Perform this task to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates received from a peer.

The MBGP translate-update feature generally is used in an MBGP-capable router that peers with a customer site that has only a BGP-capable router; the customer site has not or cannot upgrade its router to an MBGP-capable image. Because the customer site cannot originate MBGP advertisements, the router with which it peers will translate the BGP prefixes into MBGP prefixes, which are used for multicast-source Reverse Path Forwarding (RPF) lookup.

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] 	Specifies the IPv6 address family, and enters address family configuration mode.

	Command or Action	Purpose
	Example: <pre>Device(config-router)# address-family ipv6</pre>	<ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	neighbor <i>ipv6-address</i> translate-update ipv6 multicast [unicast] Example: <pre>Device(config-router-af)# neighbor 2001:DB8::2 translate-update ipv6 multicast</pre>	Generates multiprotocol IPv6 BGP updates that correspond to unicast IPv6 updates received from a peer.
Step 6	end Example: <pre>Device(config-router-af)# end</pre>	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring the IPv6 BGP Graceful Restart Capability

Procedure

	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.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65000</pre>	Enters router configuration mode for the specified routing process.
Step 4	bgp graceful-restart [<i>restart-time seconds</i> <i>stalepath-time seconds</i>] [all] Example:	Enables the BGP graceful restart capability.

	Command or Action	Purpose
	Device(config-router)# bgp graceful-restart	
Step 5	end Example: Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

Resetting IPv6 BGP Sessions

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	clear bgp ipv6 {unicast multicast} {* autonomous-system-number ip-address ipv6-address peer-group peer-group-name} [soft] [in out] Example: Device# clear bgp ipv6 unicast peer-group marketing soft out	Resets IPv6 BGP sessions.
Step 3	clear bgp ipv6 {unicast multicast} external [soft] [in out] Example: Device# clear bgp ipv6 unicast external soft in	Clears external IPv6 BGP peers.
Step 4	clear bgp ipv6 {unicast multicast} peer-group name Example: Device# clear bgp ipv6 unicast peer-group marketing	Clears all members of an IPv6 BGP peer group.
Step 5	clear bgp ipv6 {unicast multicast} dampening [ipv6-prefix/prefix-length] Example:	Clears IPv6 BGP route dampening information and unsuppresses the suppressed routes.

	Command or Action	Purpose
	Device# clear bgp ipv6 unicast dampening 2001:DB8::/64	
Step 6	clear bgp ipv6 {unicast multicast} flap-statistics [ipv6-prefix/prefix-length regexp regexp filter-list list] Example: Device# clear bgp ipv6 unicast flap-statistics filter-list 3	Clears IPv6 BGP flap statistics.

Verifying the IPv6 Multiprotocol BGP Configuration

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	show bgp ipv6 unicast multicast [ipv6-prefix/prefix-length] [longer-prefixes] [labels] Example: Device> show bgp ipv6 unicast	(Optional) Displays entries in the IPv6 BGP routing table.
Step 3	show bgp ipv6 {unicast multicast} summary Example: Device> show bgp ipv6 unicast summary	(Optional) Displays the status of all IPv6 BGP connections.
Step 4	show bgp ipv6 {unicast multicast} dampening dampened-paths Example: Device> show bgp ipv6 unicast dampening dampened-paths	(Optional) Displays IPv6 BGP dampened routes.
Step 5	debug bgp ipv6 {unicast multicast} dampening[prefix-list prefix-list-name] Example: Device# debug bgp ipv6 unicast dampening	(Optional) Displays debugging messages for IPv6 BGP dampening packets. <ul style="list-style-type: none"> • If no prefix list is specified, debugging messages for all IPv6 BGP dampening packets are displayed.

	Command or Action	Purpose
Step 6	debug bgp ipv6 unicast multicast} updates <i>[ipv6-address]</i> [prefix-list <i>prefix-list-name]</i> [in out] Example: Device# debug bgp ipv6 unicast updates	(Optional) Displays debugging messages for IPv6 BGP update packets. <ul style="list-style-type: none"> • If an <i>ipv6-address</i> argument is specified, debugging messages for IPv6 BGP updates to the specified neighbor are displayed. • Use the in keyword to display debugging messages for inbound updates only. • Use the out keyword to display debugging messages for outbound updates only.

Configuration Examples for Implementing Multiprotocol BGP for IPv6

Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer

The following example enables IPv6 globally, configures a BGP process, and establishes a BGP router ID. Also, the IPv6 multiprotocol BGP peer 2001:DB8:0:CC00::1 is configured and activated.

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# bgp router-id 192.168.99.70
Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate
Device(config-router-af)# end
```

Example: Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

The following example configures the IPv6 multiprotocol BGP peer FE80::XXXX:BFF:FE0E:A471 over Gigabit Ethernet interface 0/0/0 and sets the route map named nh6 to include the IPv6 next-hop global address of Gigabit Ethernet interface 0/0/0 in BGP updates. The IPv6 next-hop link-local address can be set by the nh6 route map (not shown in the following example) or from the interface specified by the **neighbor update-source** command (as shown in the following example).

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0111 remote-as 64600
```

```

Device(config-router)# neighbor 2001:DB8:0000:0000:0000:0000:0111 update-source
gigabitethernet 0/0/0
Device(config-router)# address-family ipv6
Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0111 activate
Device(config-router-af)# neighbor 2001:DB8:0000:0000:0000:0000:0111 route-map nh6 out
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# route-map nh6 permit 10
Device(config-route-map)# match ipv6 address prefix-list list1
Device(config-route-map)# set ipv6 next-hop 2001:DB8:5y6::1
Device(config-route-map)# exit
Device(config)# ipv6 prefix-list list1 permit 2001:DB8:2Fy2::/48 le 128
Device(config)# ipv6 prefix-list list1 deny ::/0
Device(config)# end

```



Note If you specify only the global IPv6 next-hop address (the *ipv6-address* argument) with the **set ipv6 next-hop** command after specifying the neighbor interface (the *interface-type* argument) with the **neighbor update-source** command, the link-local address of the interface specified with the *interface-type* argument is included as the next hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses.

Example Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```

Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor group1 peer-group
Device(config-router)# neighbor group1 remote-as 100
Device(config-router)# neighbor group1 update-source Loopback0
Device(config-router)# neighbor 2001:DB8::1 peer-group group1
Device(config-router)# neighbor 2001:DB8:2:2 peer-group group1
Device(config-router)# address-family ipv6 multicast
Device(config-router-af)# neighbor 2001:DB8::1 activate
Device(config-router-af)# neighbor 2001:DB8:2:2 activate
Device(config-router-af)# exit-address-family
Device(config-router)# end

```

Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

The following example configures the route map named rtp to permit IPv6 unicast routes from network 2001:DB8::/24 if they match the prefix list named list1:

```

Device> enable
Device# configure terminal
Device(config)# router bgp 64900
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64700
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate
Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 route-map rtp in
Device(config-router-af)# exit

```

```
Device(config)# ipv6 prefix-list cisco seq 10 permit 2001:DB8::/24
Device(config)# route-map rtp permit 10
Device(config-route-map)# match ipv6 address prefix-list list1
Device(config-route-map)# end
```

Example Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes BGP routes into the IPv6 multicast database of the local router:

```
router bgp 64900
  no bgp default ipv4-unicast
  address-family ipv6 multicast
  redistribute BGP
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# network 2001:DB8::/24
Device(config-router-af)# end
```

Example: Advertising IPv4 Routes Between IPv6 Peers

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. Peering is configured using IPv6 addresses in the IPv4 address family configuration mode. The inbound route map named rmap sets the next hop because the advertised next hop is likely to be unreachable.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 6peers peer-group
Device(config-router)# neighbor 2001:DB8:1234::2 remote-as 65002
Device(config-router)# address-family ipv4
Device(config-router)# neighbor 6peers activate
Device(config-router)# neighbor 6peers soft-reconfiguration inbound
Device(config-router)# neighbor 2001:DB8:1234::2 peer-group 6peers
Device(config-router)# neighbor 2001:DB8:1234::2 route-map rmap in
Device(config-router)# exit
Device(config)# route-map rmap permit 10
Device(config-route-map)# set ip next-hop 10.21.8.10
Device(config-route-map)# end
```


Additional References for Implementing Multiprotocol BGP for IPv6

Standards and RFCs

RFCs	Title
RFC 2545	<i>Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 4007	<i>IPv6 Scoped Address Architecture</i>
RFC 4364	BGP MPLS/IP Virtual Private Networks (VPNs)
RFC 4382	MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base
RFC 4659	BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>

Feature Information for Implementing Multiprotocol BGP for IPv6

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

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

Table 11: Feature Information for Implementing Multiprotocol BGP for IPv6

Feature Name	Releases	Feature Information
Multiprotocol BGP for IPv6	Cisco IOS XE Everest 16.5.1a	Multiprotocol BGP extensions for IPv6 supports the same features and functionality as IPv4 BGP.



INDEX

A

Address Resolution Protocol [54](#)
 See ARP [54](#)
 adjacency tables, with CEF [79](#)
 administrative distances [101](#)
 defined [101](#)
 ARP [62–63](#)
 encapsulation [63](#)
 static cache configuration [62](#)
 authentication keys, and routing protocols [102](#)

B

broadcast flooding [55](#)
 broadcast packets [55](#)
 directed [55](#)
 flooded [55](#)
 broadcast storms [55](#)

C

Caching Source-Active State: Example command [46](#)
 CEF [80](#)
 distributed [80](#)
 Cisco Express Forwarding [79](#)
 See CEF [79](#)
 classless routing [51](#)
 Configuration Examples for Configuring MSDP command [45](#)
 Configuring a Default MDSP Peer: Example command [45](#)
 Controlling Source Information that Your Switch Forwards: Example command [46](#)
 Controlling Source Information that Your Switch Originates: Example command [46](#)
 Controlling Source Information that Your Switch Receives: Example command [46](#)

D

default gateway [66](#)
 default networks [87](#)
 default routes [87](#)
 default routing [50](#)
 distance-vector protocols [50](#)

distribute-list command [100](#)
 dynamic routing [50](#)

E

EtherChannel [56](#)
 Layer 3 interface [56](#)

I

ICMP Router Discovery Protocol [67](#)
 See IRDP [67](#)
 inter-VLAN routing [50](#)
 IP addresses [57–58, 75](#)
 classes of [58](#)
 for IP routing [57](#)
 monitoring [75](#)
 IP broadcast address [72](#)
 IP directed broadcasts [69](#)
 IP routing [76](#)
 enabling [76](#)
 IP unicast routing [50–51, 53–58, 60, 66, 69, 72, 76, 85, 87, 89, 99, 101–102](#)
 administrative distances [101](#)
 authentication keys [102](#)
 broadcast [55, 72](#)
 address [72](#)
 flooding [55](#)
 packets [55](#)
 storms [55](#)
 classless routing [51](#)
 configuring static routes [85](#)
 default [50, 66, 87](#)
 gateways [66](#)
 networks [87](#)
 routes [87](#)
 routing [50](#)
 directed broadcasts [69](#)
 dynamic routing [50](#)
 enabling [76](#)
 EtherChannel Layer 3 interface [56](#)
 inter-VLAN [50](#)
 IP addressing [57–58](#)
 classes [58](#)

IP unicast routing (*continued*)IP addressing (*continued*)configuring [57](#)IRDP [54](#)Layer 3 interfaces [56](#)MAC address and IP address [53](#)passive interfaces [99](#)protocols [50–51](#)distance-vector [50](#)link-state [51](#)proxy ARP [54](#)redistribution [89](#)routed ports [56](#)See also RIP[IP unicast routing [51](#)zzz] [51](#)static routing [50](#)steps to configure [56](#)subnet mask [58](#)subnet zero [60](#)with SVIs [56](#)IPv6 multiprotocol BGP [105](#)IPv6 multicast address family feature [105](#)IRDP [54, 67](#)configuring [67](#)definition [54](#)**L**Layer 3 interfaces [56](#)types of [56](#)link-state protocols [51](#)**M**MAC addresses [53](#)IP address association [53](#)maximum-paths command [84](#)metric translations, between routing protocols [93](#)monitoring [75, 80](#)CEF [80](#)IP [75](#)address tables [75](#)MSDP [24](#)overview [24](#)Multiprotocol BGP [105](#)general description [105](#)**O**overview [24](#)**P**parallel paths, in routing tables [84](#)passive interfaces [99](#)configuring [99](#)PBR [95, 98](#)defined [95](#)fast-switched policy-based routing [98](#)local policy-based routing [98](#)policy-based routing [95](#)See PBR [95](#)proxy ARP [54](#)definition [54](#)with IP routing disabled [54](#)**R**Reverse Address Resolution Protocol [54](#)See RARP [54](#)route maps [95](#)policy-based routing [95](#)route-map command [97](#)routed ports [56](#)configuring [56](#)IP addresses on [56](#)routing [50, 89](#)default [50](#)dynamic [50](#)redistribution of information [89](#)static [50](#)Routing Information Protocol [51](#)See RIP [51](#)**S**static routes [85](#)configuring [85](#)static routing [50](#)subnet mask [58](#)subnet zero [60](#)SVIs [56](#)and IP unicast routing [56](#)**U**User Datagram Protocol [54, 71](#)See UDP [54, 71](#)