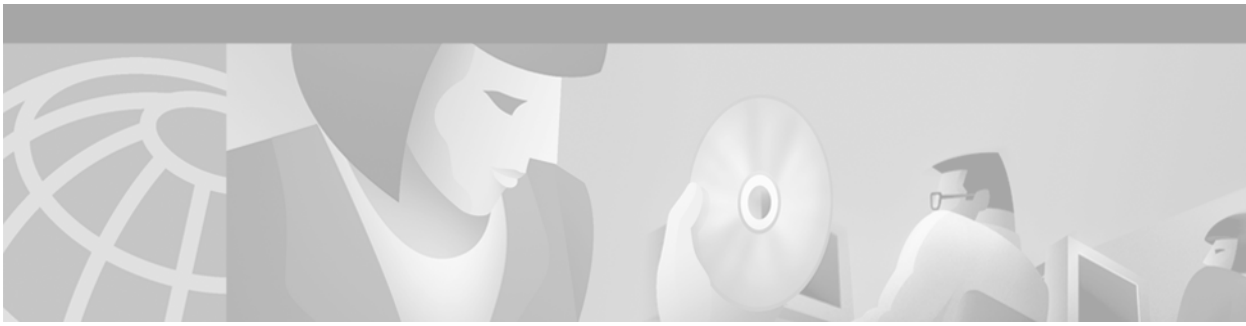


Multiprotocol BGP Extensions for IP Multicast

This feature module describes the multiprotocol Border Gateway Protocol (BGP) feature and includes the following sections:

Feature Overview

The multiprotocol BGP feature adds capabilities to BGP to enable multicast routing policy throughout the Internet and to connect multicast topologies within and between BGP autonomous systems. That is, multiprotocol BGP is an enhanced BGP that carries IP multicast routes. BGP carries two sets of routes, one set for unicast routing and one set for multicast routing. The routes associated with multicast routing are used by the Protocol Independent Multicast (PIM) to build data distribution trees.

Multiprotocol BGP is useful when you want a link dedicated to multicast traffic, perhaps to limit which resources are used for which traffic. Perhaps you want all multicast traffic exchanged at one network access point (NAP). Multiprotocol BGP allows you to have a unicast routing topology different from a multicast routing topology. Thus, you have more control over your network and resources.

In BGP, the only way to perform interdomain multicast routing was to use the BGP infrastructure that was in place for unicast routing. If those routers were not multicast capable, or there were differing policies where you wanted multicast traffic to flow, multicast routing could not be supported without multiprotocol BGP.

Note It is possible to configure BGP peers that exchange both unicast and multicast network layer reachability information (NLRI), but you cannot connect multiprotocol BGP clouds with a BGP cloud. That is, you cannot redistribute multiprotocol BGP routes into BGP.

Figure 1 illustrates a simple example of unicast and multicast topologies that are incongruent, and therefore are not be possible without multiprotocol BGP.

Autonomous systems 100, 200, and 300 are each connected to two NAPs that are FDDI rings. One is used for unicast peering (and therefore the exchanging of unicast traffic). The Multicast Friendly Interconnect (MFI) ring is used for multicast peering (and therefore the exchanging of multicast traffic). Each router is unicast- and multicast-capable.

Figure 1 Incongruent Unicast and Multicast Routes

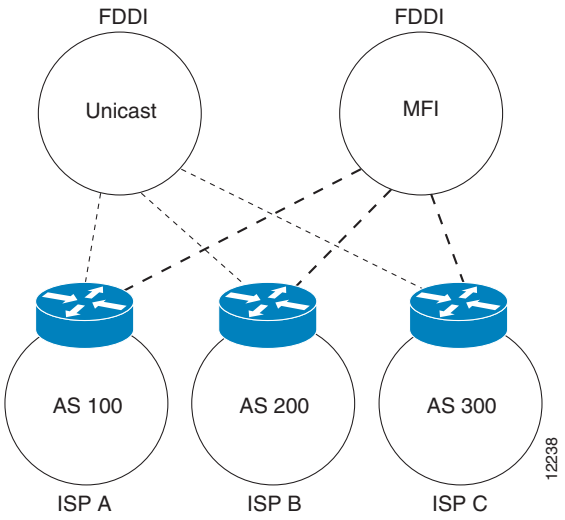


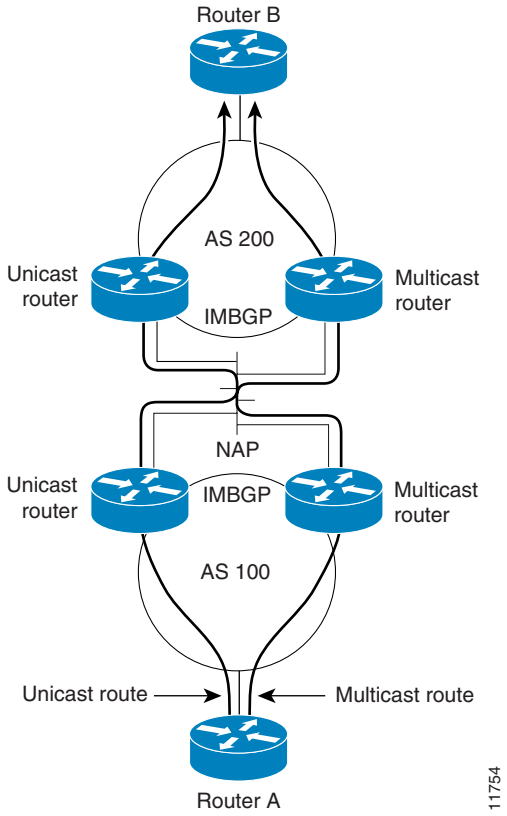
Figure 2 is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they do not support or are not configured to perform multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both unicast and multicast routing. The unicast-only and multicast-only routers are connected to a single NAP.

In Figure 2, only unicast traffic can travel from Router A to the unicast routers to Router B and back. Multicast traffic could not flow on that path, so another routing table is required. Multicast traffic uses the path from Router A to the multicast routers to Router B and back.

Figure 2 illustrates a multiprotocol BGP environment with a separate unicast route and multicast route from Router A to Router B. Multiprotocol BGP allows these routes to be noncongruent. Both of the autonomous systems must be configured for internal multiprotocol BGP.

A multicast routing protocol, such as PIM, uses the multicast BGP database to perform Reverse Path Forwarding (RPF) lookups for multicast-capable sources. Thus, packets can be sent and accepted on the multicast topology but not on the unicast topology.

Figure 2 Multicast BGP Environment



Benefits

Multiprotocol BGP offers the following benefits:

- A network can support incongruent unicast and multicast topologies.
- A network can support congruent unicast and multicast topologies that have different policies (BGP filtering configurations).
- A network can carry routing information for multiple network layer protocol address families (for example, IPv4 or VPNv4) as specified in RFC 1700, *Assigned Numbers*.
- A network that is backward compatible—routers that support the multiprotocol extensions can interoperate with routers that do not support the extensions.
- All of the routing policy capabilities of BGP can be applied to multiprotocol BGP.
- All of the BGP commands can be used with multiprotocol BGP.

Restrictions

You cannot connect multiprotocol BGP clouds with a BGP cloud. That is, you cannot redistribute multiprotocol BGP routes into BGP.

Related Features and Technologies

This feature is related to the existing BGP feature, which is documented in the *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS IP and IP Routing Command Reference*.

Related Documents

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.1
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1
- RFC 2283, *Multiprotocol Extensions for BGP-4*

Supported Platforms

This feature can run on all Cisco platforms, but it is officially supported on only the following platforms:

- Cisco 4500
- Cisco 7200
- Cisco 7500
- RSP 7000

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of MIBs supported by platform and Cisco IOS releases and to download MIB modules, go to the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2283, *Multiprotocol Extensions for BGP-4*

Prerequisites

This document assumes you are familiar with BGP and IP multicast routing. For more information, refer to the “Configuring BGP” and “Configuring IP Multicast Routing” chapters of the *Cisco IOS IP and IP Routing Configuration Guide*.

Configuration Tasks

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional:

- Understanding NLRI Keywords and Address Families (Required)
- Configuring a Multiprotocol BGP Peer (Required)
- Configuring a Multiprotocol BGP Peer Group (Optional)
- Advertising Routes into Multiprotocol BGP (Required)
- Configuring Route Maps for Multiprotocol BGP Prefixes (Optional)
- Redistributing Prefixes into Multiprotocol BGP (Required)
- Configuring DVMRP Interoperability with Multiprotocol BGP (Optional)
- Configuring a Multiprotocol BGP Route Reflector (Optional)
- Configuring Aggregate Multiprotocol BGP Addresses (Optional)
- Verifying Multiprotocol BGP Configuration and Operation (Optional)

Understanding NLRI Keywords and Address Families

Multiprotocol BGP was introduced in Cisco IOS Release 11.1(20)CC and Cisco IOS Release 12.0(2)S prior to it being integrated into Cisco IOS Release 12.0(7)T. In Cisco IOS Release 11.1(20)CC and later releases and Cisco IOS Release 12.0(2)S and later releases, the Cisco IOS software uses NLRI keywords to enable multiprotocol BGP over a BGP session and to populate unicast BGP prefixes in the unicast database and multicast BGP prefixes in the multicast database. In Cisco IOS Release 12.0(7)T, the Cisco IOS software uses separate address families to enable multiprotocol BGP over a BGP session and to populate unicast BGP prefixes in the unicast database and multicast BGP prefixes in the multicast database.

Cisco IOS Release 12.0(7)T does not support the NLRI keywords. However, for backward compatibility, the NLRI keyword configuration of a Cisco router is automatically converted to an address family configuration when a router is upgraded to Cisco IOS Release 12.0(7)T. The following example shows an NLRI keyword configuration for a Cisco router that is running Cisco IOS Release 12.0(8)S:

```
router bgp 5
  no synchronization
  network 172.16.214.0 mask 255.255.255.0 nlri unicast multicast
  neighbor 172.16.214.34 remote-as 5
  neighbor 172.16.214.38 remote-as 2 nlri unicast multicast
  neighbor 172.16.214.42 remote-as 5
  neighbor 172.16.214.59 remote-as 5
no auto-summary
```

The following example shows the resulting address family configuration after the same router is upgraded to Cisco IOS Release 12.0(7)T:

```
router bgp 5
  no synchronization
  network 172.16.214.0 mask 255.255.255.0
  neighbor 172.16.214.34 remote-as 5
  neighbor 172.16.214.38 remote-as 2
  neighbor 172.16.214.42 remote-as 5
  neighbor 172.16.214.59 remote-as 5
no auto-summary
```

Note Although supported in Cisco IOS Release 12.0(7)T, this feature module does not explain how to configure the BGP-4 extensions for Virtual Private Network (VPN) address family prefixes. Configuring VPN address family prefixes will be explained in a later release of the *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS IP and IP Routing Command Reference*.

Configuring a Multiprotocol BGP Peer

To configure multiprotocol BGP between two routers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# neighbor ip-address remote-as autonomous-system-number`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router.
Step 3	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies the IPv4 address family type and places the router in address family configuration mode.
Step 4	`Router(config-router-af)# neighbor {ip-address | peer-group-name} activate`	Enables the neighbor to exchange prefixes for the specified family type with the local router.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To exchange other address prefix types, such as multicast and VPNv4, neighbors must also be activated using the **neighbor activate** command in address family configuration mode, as shown.

See the "Multiprotocol BGP Peer Examples" section for multiprotocol BGP peer configuration examples.

Configuring a Multiprotocol BGP Peer Group

To configure a peer group to perform multiprotocol BGP routing, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# neighbor peer-group-name peer-group`	Creates a multiprotocol BGP peer group.
Step 3	`Router(config-router)# neighbor ip-address remote-as autonomous-system-number`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router.
Step 4	`Router(config-router)# neighbor ip-address peer-group peer-group-name`	Assigns the IP address of a BGP neighbor to a peer group.
Step 5	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies IPv4 address family type and places the router in address family configuration mode.
Step 6	`Router(config-router-af)# neighbor peer-group-name activate`	Enables the peer group to exchange prefixes for the specified family type with the neighbor and the local router.
Step 7	`Router(config-router-af)# neighbor ip-address peer-group peer-group-name`	Assigns the IP address of a BGP neighbor to a peer group.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To exchange other address prefix types, such as multicast and VPNv4, neighbors must also be activated using the **neighbor activate** command in address family configuration mode, as shown.

Note Peer groups that are defined in router configuration mode using the **neighbor peer-group** command exchange only unicast address prefixes by default. To exchange other address prefix types, such as multicast, peer groups must be defined in address family configuration mode using the **neighbor activate** command, as shown.

Members of a peer group automatically inherit the address prefix configuration of the peer group.

Refer to the section "Configure BGP Peer Groups" of the "Configuring BGP" chapter in the *Cisco IOS IP and IP Routing Configuration Guide* for information and instructions on assigning options to the peer group and making a BGP or multiprotocol BGP neighbor a member of the peer group. See the "Multiprotocol BGP Peer Group Examples" section for multiprotocol BGP peer group configuration examples.

Advertising Routes into Multiprotocol BGP

To advertise (inject) a network number and mask into multiprotocol BGP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies IPv4 address family type and places the router in address family configuration mode.
Step 3	`Router(config-router-af)# network network-number [mask network-mask]`	Advertises (injects) this network number and mask into the multicast BGP database. (The routes must first be found in the unicast forwarding table.) Specifically, the network number and mask are injected into the multicast database for the address family specified in the previous step. Routes are tagged from the specified network as "local origin."

Note Networks that are defined in router configuration mode using the **network** command are injected into the unicast database by default. To inject a network into another database, such as the multicast database, the network must be defined in address family configuration mode using the **network** command, as shown.

See the "Multiprotocol BGP Network Advertisement Examples" section for multiprotocol BGP network advertisement configuration examples.

Configuring Route Maps for Multiprotocol BGP Prefixes

To configure a route map for multiprotocol BGP prefixes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# neighbor ip-address remote-as autonomous-system-number`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router.
Step 3	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies IPv4 address family type and places the router in address family configuration mode.
Step 4	`Router(config-router-af)# neighbor ip-address activate`	Enables the address family for the neighbor in the remote autonomous system.
Step 5	`Router(config-router-af)# neighbor ip-address route-map route-map-name {in | out}`	Applies a route map to incoming or outgoing routes.

	Command	Purpose
Step 6	`Router(config)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map.
Step 7	`Router(config-route-map)# match ip-address access-list-number`	Distributes any routes that have a destination network number address permitted by a standard or extended access list, or performs policy routing on packets.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To exchange other address prefix types, such as multicast and VPNv4, neighbors must also be activated using the **neighbor activate** command in address family configuration mode, as shown.

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

See the "Multiprotocol BGP Route Map Examples" section for multiprotocol BGP route map configuration examples.

Redistributing Prefixes into Multiprotocol BGP

Redistribution is the process of injecting prefixes from one routing protocol into another routing protocol. The tasks in this section explain how to inject prefixes from a routing protocol into multiprotocol BGP. Specifically, prefixes that are redistributed into multiprotocol BGP using the **redistribute** command are injected into the unicast database, the multicast database, or both.

To inject prefixes from a routing protocol into multiprotocol BGP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies IPv4 address family type and places the router in address family configuration mode.
Step 3	`Router(config-router-af)# redistribute protocol [process-id] [route-map map-name]`	Specifies the routing protocol from which prefixes should be redistributed into multiprotocol BGP.

	Command	Purpose
Step 4	`Router(config)# route-map map-tag [permit | deny] [sequence-number]` `Router(config-route-map)#`	Defines a route map and places the router in route map configuration mode. Follow this step with a **match** command.
Step 5	`Router(config-route-map)# match ip-address access-list-number`	Distributes any prefixes that have a destination network number address permitted by a standard or extended access list, or performs policy routing on packets.

Note Route maps that are applied in router configuration mode using the **redistribute route-map** command are applied to unicast address prefixes by default. Route maps for other address families, such as multicast, must be applied in address family configuration mode using the **redistribute route-map** command, as shown.

See the "Multiprotocol BGP Route Redistribution Examples" section for multiprotocol BGP route redistribution configuration examples.

Configuring DVMRP Interoperability with Multiprotocol BGP

Cisco multicast routers using PIM can interoperate with non-Cisco multicast routers that use the Distance Vector Multicast Routing Protocol (DVMRP).

PIM routers dynamically discover DVMRP multicast routers on attached networks. Once a DVMRP neighbor has been discovered, the router caches DVMRP routes that the neighbor sends. Those routes describe sources in a DVMRP cloud that want their packets to be received by receivers outside of this routing domain. Multiprotocol BGP allows the source prefixes of those sources to be known outside of the routing domain.

The router periodically sends DVMRP Report messages advertising the unicast sources reachable in the PIM domain.

Redistributing Multiprotocol BGP Routes into DVMRP

By default, no multiprotocol BGP routes are redistributed into DVMRP. However, you can configure all multiprotocol BGP routes to be redistributed into DVMRP with a specified metric. Furthermore, to redistribute only certain multiprotocol BGP routes into DVMRP, you can configure the metric and subject it to route map conditions. If you supply a route map, you can specify various match criteria options for the multiprotocol BGP routes. If the route passes the route map, then the route is redistributed into DVMRP.

If there are multicast sources in other routing domains that are known via multiprotocol BGP and there are receivers in a DVMRP cloud, they will want to receive packets from those sources. Therefore, you need to redistribute the multiprotocol BGP prefix routes into DVMRP. This will be the scenario when distributing multiprotocol BGP prefixes into the MBONE.

To redistribute multiprotocol BGP routes into DVMRP, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip dvmrp metric` *metric* `[route-map` *map-name*`] mbgp`	Redistributes multiprotocol BGP routes into DVMRP with a specified metric. An optional route map controls which routes are redistributed; otherwise, all multiprotocol BGP routes are redistributed.

Redistributing DVMRP Routes into Multiprotocol BGP

If there are multicast sources in a DVMRP routing domain that need to reach receivers in multiprotocol BGP routing domains, you need to redistribute DVMRP prefixes into multiprotocol BGP. If you supply a route map, you can also use the **set** commands to specify various BGP attribute settings.

To redistribute DVMRP prefixes into multiprotocol BGP, use the following command in address family configuration mode:

Command	Purpose
`Router(config-router-af)# redistribute dvmrp [route-map` *map-name*`]`	Redistributes DVMRP routes into multiprotocol BGP.

To redistribute DVMRP prefixes into multiprotocol BGP, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# redistribute dvmrp [route-map` *map-name*`]`	Redistributes DVMRP routes into multiprotocol BGP.

See the "Multiprotocol BGP Route Redistribution Examples" section for multiprotocol BGP route redistribution configuration examples.

Configuring a Multiprotocol BGP Route Reflector

To configure a local router as a route reflector of multiprotocol BGP prefixes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp` *autonomous-system* `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# neighbor` *ip-address* `remote-as` *autonomous-system-number*	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router.
Step 3	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies an IPv4 address family type and places the router in address family configuration mode.

	Command	Purpose
Step 4	`Router(config-router-af)# neighbor ip-address activate`	Enables the specified address family for the neighbor in the remote autonomous system.
Step 5	`Router(config-router-af)# neighbor ip-address route-reflector-client`	Configures the router as a route reflector of prefixes for the specified address family type and configures the specified neighbor as its client.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To exchange other address prefix types, such as multicast and VPNv4, neighbors must also be activated using the **neighbor activate** command in address family configuration mode, as shown.

Note Route reflectors and clients (neighbors or internal BGP peer groups) that are defined in router configuration mode using the **neighbor route-reflector-client** command reflect unicast address prefixes to and from those clients by default. To reflect prefixes for other address families, such as multicast, define the reflectors and clients in address family configuration mode using the **neighbor route-reflector-client** command, as shown.

See the "Multiprotocol BGP Route Reflector Examples" section for multiprotocol BGP route reflector configuration examples.

Configuring Aggregate Multiprotocol BGP Addresses

The tasks in this section explain how to configure an aggregate address for multiprotocol BGP. Specifically, the tasks in this section explain how to inject an aggregate address into the multicast database, the unicast database, or both.

To configure an aggregate address for multiprotocol BGP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router bgp autonomous-system` `Router(config-router)#`	Configures a BGP routing process and places the router in router configuration mode.
Step 2	`Router(config-router)# address-family ipv4 multicast` `Router(config-router-af)#`	Specifies IPv4 address family type and places the router in address family configuration mode.
Step 3	`Router(config-router-af)# aggregate-address address mask [as-set] [summary-only] [suppress-map map-name] [advertise-map map-name] [attribute-map map-name]`	Configures an aggregate address with various options.

Note Aggregate addresses that are defined in router configuration mode using the **aggregate-address as-set** command are injected into the unicast database by default. To enter an aggregate address in another database, such as the multicast database, the aggregate address must be defined in address family configuration mode using the **aggregate-address as-set** command, as shown.

See the "Aggregate Multiprotocol BGP Address Examples" section for aggregate multiprotocol BGP address configuration examples.

Verifying Multiprotocol BGP Configuration and Operation

The following steps show an example of how to verify multiprotocol BGP configuration and operation:

Step 1 Enter the **show ip bgp ipv4 multicast** EXEC command to display information related to the multicast database:

```
Router# show ip bgp ipv4 multicast

MBGP table version is 6, local router ID is 192.168.200.66
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
*> 10.0.20.16/28     0.0.0.0                0      0 32768 i
*> 10.0.35.16/28     0.0.0.0                0      0 32768 i
*> 10.0.36.0/28      0.0.0.0                0      0 32768 i
*> 10.0.48.16/28     0.0.0.0                0      0 32768 i
*> 10.2.0.0/16       0.0.0.0                0      0 32768 i
*> 10.2.1.0/24       0.0.0.0                0      0 32768 i
*> 10.2.2.0/24       0.0.0.0                0      0 32768 i
*> 10.2.3.0/24       0.0.0.0                0      0 32768 i
*> 10.2.7.0/24       0.0.0.0                0      0 32768 i
*> 10.2.8.0/24       0.0.0.0                0      0 32768 i
*> 10.2.10.0/24      0.0.0.0                0      0 32768 i
*> 10.2.11.0/24      0.0.0.0                0      0 32768 i
*> 10.2.12.0/24      0.0.0.0                0      0 32768 i
*> 10.2.13.0/24      0.0.0.0                0      0 32768 i
```

Note For a description of each output display field, refer to the **show ip bgp ipv4 multicast** command in the "Command Reference" section of this feature module.

Step 2 Enter the **show ip bgp ipv4 multicast summary** EXEC command to display a summary of multicast database information:

```
Router# show ip bgp ipv4 multicast summary

BGP router identifier 10.0.33.34, local AS number 34
BGP table version is 5, main routing table version 1
4 network entries and 6 paths using 604 bytes of memory
5 BGP path attribute entries using 260 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP community entries using 48 bytes of memory
2 BGP route-map cache entries using 32 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 8/28 prefixes, 12/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.33.35      4    35     624     624        5    0    0 10:13:46        3
```

Step 3 Enter the **debug ip mbgp dampening** EXEC command to log the route flap dampening activity:

```
Router# debug ip mbgp dampening

BGP: charge penalty for 173.19.0.0/16 path 49 with halflife-time 15 reuse/suppress
750/2000
BGP: flapped 1 times since 00:00:00. New penalty is 1000
BGP: charge penalty for 173.19.0.0/16 path 19 49 with halflife-time 15 reuse/suppress
750/2000
BGP: flapped 1 times since 00:00:00. New penalty is 1000
```

Step 4 Enter the **debug ip mbgp updates** EXEC command to log the multiprotocol BGP-related information passed in BGP Update messages:

```
Router# debug ip mbgp updates

BGP: NEXT_HOP part 1 net 200.10.202.0/24, neigh 171.69.233.49, next 171.69.233.34
BGP: 171.69.233.49 send UPDATE 200.10.202.0/24, next 171.69.233.34, metric 0, path 33 34
19 49 109 65000 297 1239 1800 3597
BGP: NEXT_HOP part 1 net 200.10.228.0/22, neigh 171.69.233.49, next 171.69.233.34
BGP: 171.69.233.49 rcv UPDATE about 222.2.2.0/24, next hop 171.69.233.49, path 49 109
metric 0
BGP: 171.69.233.49 rcv UPDATE about 131.103.0.0/16, next hop 171.69.233.49, path 49 109
metric 0
BGP: 171.69.233.49 rcv UPDATE about 206.205.242.0/24, next hop 171.69.233.49, path 49 109
metric 0
```

Step 5 Enter the **show ip mpacket quality** EXEC command to display the quality of Real-Time Transport Protocol (RTP) data based on packets captured in the IP multicast cache header buffer:

```
Router# show ip mpacket 224.2.163.188 quality

Calculating RTP data quality for 224.2.163.188
  Session: UO Presents KKNU New Country
  Source: 128.223.83.27 (sand.uoregon.edu), Port: 23824
    Packets received: 83, lost: 5, loss percentage: 5.6%
    Packets misordered: 7, average loss gap: 0
```

Configuration Examples

This section provides the following multiprotocol BGP configuration examples:

- Multiprotocol BGP Peer Examples
- Multiprotocol BGP Peer Group Examples
- Multiprotocol BGP Network Advertisement Examples
- Multiprotocol BGP Route Map Examples
- Multiprotocol BGP Route Redistribution Examples
- Multiprotocol BGP Route Reflector Examples
- Aggregate Multiprotocol BGP Address Examples

Multiprotocol BGP Peer Examples

The following example shows how to use an address family to configure a neighbor as both unicast and multicast capable:

```
router bgp 100
address-family ipv4 unicast
  neighbor 10.1.1.1 activate

router bgp 100
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
```

Multiprotocol BGP Peer Group Examples

The following example shows how to use an address family to configure a peer group so that all members of the peer group are both unicast and multicast capable:

```
router bgp 100
neighbor 10.1.1.1 remote-as 1
neighbor 12.2.2.2 remote-as 2
address-family ipv4 unicast
  neighbor mygroup peer-group
  neighbor 10.1.1.1 peer-group mygroup
  neighbor 12.2.2.2 peer-group mygroup

router bgp 100
neighbor 10.1.1.1 remote-as 1
neighbor 12.2.2.2 remote-as 2
address-family ipv4 multicast
  neighbor mygroup peer-group
  neighbor 10.1.1.1 peer-group mygroup
  neighbor 12.2.2.2 peer-group mygroup
  neighbor 10.1.1.1 activate
  neighbor 12.2.2.2 activate
```

Multiprotocol BGP Network Advertisement Examples

The following examples show how to use an address family to inject a network number and mask into the unicast database and the multicast database:

```
router bgp 100
 address-family ipv4 unicast
  neighbor 10.0.0.0  255.0.0.0

router bgp 100
 address-family ipv4 multicast
  neighbor 10.0.0.0  255.0.0.0
```

Multiprotocol BGP Route Map Examples

The following example shows how to use an address family to configure BGP so that any unicast and multicast routes from neighbor 10.1.1.1 are accepted if they match access list 1:

```
router bgp 109
neighbor 10.1.1.1 remote-as 1
address-family ipv4 unicast
 neighbor 10.1.1.1 route-map filter-some-multicast in

router bgp 109
neighbor 10.1.1.1 remote-as 1
address-family ipv4 multicast
 neighbor 10.1.1.1 route-map filter-some-multicast in
 neighbor 10.1.1.1 activate

route-map filter-some-multicast
 match ip address 1
```

Multiprotocol BGP Route Redistribution Examples

The following example shows how to use an address family to redistribute DVMRP routes that match access list 1 into the multicast database and the unicast database of the local router:

```
router bgp 109
address-family ipv4 unicast
 redistribute dvmrp route-map dvmrp-into-mbgp

router bgp 109
address-family ipv4 multicast
 redistribute dvmrp route-map dvmrp-into-mbgp

route-map dvmrp-into-mbgp
 match ip address 1
```

Multiprotocol BGP Route Reflector Examples

The following example show how to use an address family to configure internal BGP peer 10.1.1.1 as a route-reflector client for both unicast and multicast prefixes:

```
router bgp 109
 address-family ipv4 unicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client

router bgp 109
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client
```

Aggregate Multiprotocol BGP Address Examples

The following example show how to use an address family to configure an aggregate multiprotocol BGP address entry in both the unicast database and the multicast database:

```
router bgp 109
 address-family ipv4 unicast
  aggregate-address 172.16.0.0 255.0.0.0 as-set

router bgp 109
 address-family ipv4 multicast
  aggregate-address 172.16.0.0 255.0.0.0 as-set
```

Command Reference

This section documents new or modified commands. All other commands used with multiprotocol BGP are documented in the Cisco IOS Release 12.1 command reference publications.

- **address-family ipv4**
- **address-family vpnv4**
- **aggregate-address**
- **distance bgp**
- **ip dvmrp metric**
- **ip multicast cache-headers**
- **neighbor peer-group (creating)**
- **neighbor remote-as**
- **neighbor route-map**
- **network (BGP and multiprotocol BGP)**
- **redistribute dvmrp**
- **show ip bgp ipv4 multicast**
- **show ip bgp ipv4 multicast summary**

Commands in this feature module that have been replaced by new or existing commands are no longer documented. Table 1 maps the old commands with their replacements.

Table 1 Mapping Old Commands with Replacement Commands

Old Command	Replacement Command
distance mbgp	**distance bgp**
match nlri	**address-family ipv4** **address-family vpnv4**
set nlri	**address-family ipv4** **address-family vpnv4**
show ip mbgp	**show ip bgp ipv4 multicast**
show ip mbgp summary	**show ip bgp ipv4 multicast summary**

address-family ipv4

To enter address family configuration mode for configuring routing sessions, such as Border Gateway Protocol (BGP), that use standard IPv4 address prefixes, use the **address-family ipv4** router configuration command. To disable address family configuration mode, use the **no** form of this command.

address-family ipv4 [**multicast** | **unicast** [**vrf** *vrf-name*]

no address-family ipv4 [**multicast** | **unicast** | **vrf** *vrf-name*]

Syntax Description

multicast	(Optional) Specifies IPv4 multicast address prefixes.
unicast	(Optional) Specifies IPv4 unicast address prefixes.
vrf *vrf-name*	(Optional) Specifies the name of the virtual routing/forwarding instance (VRF) to associate with subsequent IPv4 address family configuration mode commands.

Defaults

IPv4 address prefixes are not enabled. Unicast address prefixes are the default when IPv4 address prefixes are configured.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The **address-family ipv4** command places the router in address family configuration mode (prompt: `config-router-af`), from which you can configure routing sessions that use standard IPv4 address prefixes. To leave address family configuration mode and return to router configuration mode, type **exit**.

Routing information for address family IPv4 is advertised by default when you configure a BGP routing session using the **neighbor remote-as** command unless you execute the **no bgp default ipv4-unicast** command.

The **address-family ipv4** command replaces the **match nlri** and **set nlri** commands.

Examples

The following example places the router in address family configuration mode for the IPv4 address family:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4
Router(config-router-af)#
```

The following example places the router in address family configuration mode and specifies multicast address prefixes for the IPv4 address family:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 multicast
Router(config-router-af)#
```

The following example places the router in address family configuration mode and specifies unicast address prefixes for the IPv4 address family:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)#
```

The following example places the router in address family configuration mode and specifies cisco as the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf cisco
Router(config-router-af)#
```

Use this form of the command, which specifies a VRF, only to configure routing exchanges between provider edge (PE) and customer edge (CE) devices.

Related Commands

Command	Description
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPNv4 address prefixes.
neighbor activate	Enables the exchange of information with a BGP neighboring router.

address-family vpnv4

To enter address family configuration mode for configuring routing sessions, such as Border Gateway Protocol (BGP), that use standard VPNv4 address prefixes, use the **address-family vpnv4** router configuration command. To disable address family configuration mode, use the **no** form of this command.

address-family vpnv4 [**unicast**]

no address-family vpnv4 [**unicast**]

Syntax Description

unicast	(Optional) Specifies VPNv4 unicast address prefixes.

Defaults

VPNv4 address prefixes are not enabled. Unicast address prefixes are the default when VPNv4 address prefixes are configured.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The **address-family vpnv4** command places the router in address family configuration mode (prompt: `config-router-af`), from which you can configure routing sessions that use VPNv4 address prefixes. To leave address family configuration mode and return to router configuration mode, type **exit**.

The **address-family vpnv4** command replaces the **match nlri** and **set nlri** commands.

Examples

The following example places the router in address family configuration mode for the VPNv4 address family:

```
Router(config)# router bgp 100
(config-router)# address-family vpnv4
(config-router-af)#
```

The following example places the router in address family configuration mode for the unicast VPNv4 address family:

```
Router(config)# router bgp 100
(config-router)# address-family vpnv4 unicast
(config-router-af)#
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.
neighbor activate	Enables the exchange of information with a BGP neighboring router.

aggregate-address

To create an aggregate entry in a Border Gateway Protocol (BGP) or multicast BGP database, use the **aggregate-address** command in address family or router configuration mode. To disable this function, use the **no** form of this command.

aggregate-address *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

no aggregate-address *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

Syntax Description

address	Aggregate address.
mask	Aggregate mask.
as-set	(Optional) Generates autonomous system set path information.
summary-only	(Optional) Filters all more-specific routes from updates.
suppress-map *map-name*	(Optional) Name of the route map used to select the routes to be suppressed.
advertise-map *map-name*	(Optional) Name of the route map used to select the routes to create AS-SET origin communities.
attribute-map *map-name*	(Optional) Name of route map used to set the attribute of the aggregate route.

Defaults

Disabled

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
10.0	This command was introduced.
11.1(20)CC	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(2)S	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(7)T	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were removed. Address family configuration mode was added.

Usage Guidelines

You can implement aggregate routing in BGP and multiprotocol BGP either by redistributing an aggregate route into BGP or multiprotocol BGP, or by using this conditional aggregate routing feature.

Using the **aggregate-address** command with no arguments will create an aggregate entry in the BGP or multicast BGP database if there are any more-specific BGP or multiprotocol BGP routes available that fall in the specified range. The aggregate route will be advertised as coming from your autonomous system and will have the atomic aggregate attribute set to show that information might be missing. (By default, the atomic aggregate attribute is set unless you specify the **as-set** keyword.)

Using the **as-set** keyword creates an aggregate entry using the same rules that the command follows without this keyword, but the path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized. Do not use this form of the **aggregate-address** command when aggregating many paths, because this route must be continually withdrawn and reupdated as autonomous system path reachability information for the summarized routes changes.

Using the **summary-only** keyword not only creates the aggregate route (for example, 193.*.*.*) but also suppresses advertisements of more-specific routes to all neighbors. If you only want to suppress advertisements to certain neighbors, you may use the **neighbor distribute-list** command, with caution. If a more-specific route leaks out, all BGP or multiprotocol BGP routers will prefer that route over the less-specific aggregate you are generating (using longest-match routing).

Using the **suppress-map** keyword creates the aggregate route but suppresses advertisement of specified routes. You can use the **match** clauses of route maps to selectively suppress some more-specific routes of the aggregate and leave others unsuppressed. IP access lists and autonomous system path access lists match clauses are supported.

Examples

In the following example, a BGP aggregate address is created in router configuration mode. The path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized.

```
Router(config)# router bgp 5
Router(config-router)# aggregate-address 193.0.0.0 255.0.0.0 as-set
```

In the following example, a multiprotocol BGP aggregate address is created in address family configuration mode and applied to the multicast database only using an IPv4 address family. More-specific routes are filtered from updates.

```
Router(config)# router bgp 5
Router(config-router)# address-family ipv4 multicast
Router(config-router-af)# aggregate-address 193.0.0.0 255.0.0.0 summary-only
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.
match ip address	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on packets.
route-map (IP)	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.

distance bgp

To allow the use of external, internal, and local administrative distances that could be a better route than other external, internal, or local routes to a node, use the **distance bgp** command in address family or router configuration mode. To return to the default values, use the **no** form of this command.

distance bgp *external-distance internal-distance local-distance*

no distance bgp

Syntax Description

external-distance	Administrative distance for Border Gateway Protocol (BGP) external routes. External routes are routes for which the best path is learned from a neighbor external to the autonomous system. Acceptable values are from 1 to 255. The default is 20. Routes with a distance of 255 are not installed in the routing table.
internal-distance	Administrative distance for BGP internal routes. Internal routes are those routes that are learned from another BGP entity within the same autonomous system. Acceptable values are from 1 to 255. The default is 200. Routes with a distance of 255 are not installed in the routing table.
local-distance	Administrative distance for BGP local routes. Local routes are those networks listed with a **network** router configuration command, often as back doors, for that router or for networks that are being redistributed from another process. Acceptable values are from 1 to 255. The default is 200. Routes with a distance of 255 are not installed in the routing table.

Defaults

external-distance: 20
internal-distance: 200
local-distance: 200

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(7)T	Address family configuration mode was added.

Usage Guidelines

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is a positive integer from 1 to 255. In general, the higher the value, the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored.

Use this command if another protocol is known to be able to provide a better route to a node than was actually learned via external BGP, or if some internal routes should be preferred by the BGP.

Caution Changing the administrative distance of BGP internal routes is considered dangerous and is not recommended. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.

The **distance bgp** command replaces the **distance mbgp** command.

Examples

In the following router configuration mode example, internal routes are known to be preferable to those learned through the Interior Gateway Protocol (IGP), so the administrative distance values are set accordingly:

```
router bgp 109
 network 131.108.0.0
 neighbor 129.140.6.6 remote-as 123
 neighbor 128.125.1.1 remote-as 47
 distance bgp 20 20 200
```

In the following address family configuration mode example, internal routes are known to be preferable to those learned through IGP, so the administrative distance values are set accordingly:

```
router bgp 109
neighbor 129.140.6.6 remote-as 123
neighbor 128.125.1.1 remote-as 47
address family ipv4 multicast
 network 131.108.0.0
 distance bgp 20 20 200
 neighbor 129.140.6.6 activate
 neighbor 128.125.1.1 activate
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.

distance mbgp

In Cisco IOS Release 12.0(7)T, the **distance bgp** command replaces the **distance mbgp** command. See the description of the **distance bgp** command for more information.

ip dvmrp metric

To configure the metric associated with a set of destinations for Distance Vector Multicast Routing Protocol (DVMRP) reports, use the **ip dvmrp metric** interface configuration command. (Note that this command has two different syntax possibilities.) To disable this function, use the **no** form of this command.

ip dvmrp metric *metric* [**route-map** *map-name*] [**mbgp**] [**list** *access-list-number*] [[*protocol process-id*] | **dvmrp**]

no ip dvmrp metric *metric* [**route-map** *map-name*] [**mbgp**] [**list** *access-list-number*] [[*protocol process-id*] | **dvmrp**]

Syntax Description

metric	Metric associated with a set of destinations for DVMRP reports. It can be a value from 0 to 32. A value of 0 means that the route is not advertised. A value of 32 is equivalent to infinity (unreachable).
route-map *map-name*	(Optional) Name of a route map. If you specify this argument, only the destinations that match the route map are reported with the configured metric. Unicast routes are subject to route map conditions before being injected into DVMRP. Route maps cannot be used for DVMRP routes.
mbgp	(Optional) Configures redistribution of only IPv4 multicast routes into DVMRP.
list *access-list-number*	(Optional) Number of an access list. If you specify this argument, only the multicast destinations that match the access list are reported with the configured metric. Any destinations not advertised because of split horizon do not use the configured metric.
protocol	(Optional) Name of unicast routing protocol, such as **bgp**, **dvmrp**, **eigrp**, **igrp**, **isis**, **ospf**, **rip**, or **static**. If you specify these values, only routes learned by the specified routing protocol are advertised in DVMRP report messages.
process-id	(Optional) Process ID number of the unicast routing protocol.
dvmrp	(Optional) Allows routes from the DVMRP routing table to be advertised with the configured *metric*, or filtered.

Defaults

No metric is preconfigured. Only directly connected subnets and networks are advertised to neighboring DVMRP routers.

Command Modes

Interface configuration

Command History

Release	Modification
10.2	This command was introduced.
11.1	The **route-map** keyword was added.
11.1(20)CC	This **mbgp** keyword was added.
12.0(7)T	This **mbgp** keyword was added.

Usage Guidelines

When Protocol Independent Multicast (PIM) is configured on an interface and DVMRP neighbors are discovered, the Cisco IOS software sends DVMRP report messages for directly connected networks. The **ip dvmrp metric** command enables DVMRP report messages for multicast destinations that match the access list. Usually, the metric for these routes is 1. Under certain circumstances, you might want to tailor the metric used for various unicast routes. This command lets you configure the metric associated with a set of destinations for Report messages sent out this interface.

You can use the *access-list-number* argument in conjunction with the *protocol process-id* arguments to selectively list the destinations learned from a given routing protocol.

To display DVMRP activity, use the **debug ip dvmrp** command.

Examples

The following example connects a PIM cloud to a DVMRP cloud. Access list 1 permits the sending of DVMRP reports to the DVMRP routers advertising all sources in the 198.92.35.0 network with a metric of 1. Access list 2 permits all other destinations, but the metric of 0 means that no DVMRP reports are sent for these destinations.

```
access-list 1 permit 198.92.35.0 0.0.0.255
access-list 1 deny 0.0.0.0 255.255.255.255
access-list 2 permit 0.0.0.0 255.255.255.255
interface tunnel 0
 ip dvmrp metric 1 list 1
 ip dvmrp metric 0 list 2
```

The following example redistributes IPv4 multicast routes into DVMRP neighbors with a metric of 1:

```
interface tunnel 0
 ip dvmrp metric 1 mbgp
```

Related Commands

Command	Description
debug ip dvmrp	Displays information on DVMRP packets received and sent.
ip dvmrp accept-filter	Configures an acceptance filter for incoming DVMRP reports.

ip multicast cache-headers

To allocate a circular buffer to store IPv4 multicast packet headers that the router receives, use the **ip multicast cache-headers** global configuration command. To disable the buffer, use the **no** form of this command.

ip multicast cache-headers [**rtp**]

no ip multicast cache-headers

Syntax Description

rtp	(Optional) Caches Real-Time Transport Protocol (RTP) headers.

Defaults

Disabled

Command Modes

Global configuration

Command History

Release	Modification
11.1	This command was introduced.
11.1(20)CC	The **rtp** keyword was added.
12.0(7)T	The **rtp** keyword was added.

Usage Guidelines

You can store IPv4 multicast packet headers in a cache and then display them to determine the following:

- Who is sending IP multicast packets to which groups
- Interpacket delay
- Duplicate IP multicast packets (if any)
- Multicast forwarding loops in your network (if any)
- Scope of the group
- User Datagram Protocol (UDP) port numbers
- Packet length

Note This feature allocates a circular buffer of approximately 32 KB. Do not configure this feature if the router is low on memory.

Use the **show ip mpacket** command to display the buffer.

Examples

The following example allocates a buffer to store IPv4 multicast packet headers:

```
ip multicast cache-headers
```

Related Commands

Command	Description
show ip mpacket	Displays the contents of the circular cache-header buffer.
show ip mpacket quality	Displays RTP data quality based on packets captured in the IP multicast cache header buffer.

match nlri

In Cisco IOS Release 12.0(7)T, the **address-family ipv4** and **address-family vpnv4** commands replace the **match nlri** command. See the descriptions of the **address-family ipv4** or **address-family vpnv4** command for more information.

neighbor peer-group (creating)

To create a Border Gateway Protocol (BGP) or multiprotocol BGP peer group, use the **neighbor peer-group** command in address family or router configuration mode. To remove the peer group and all of its members, use the **no** form of this command.

neighbor *peer-group-name* **peer-group**

no neighbor *peer-group-name* **peer-group**

Syntax Description

peer-group-name	Name of the BGP peer group.

Defaults

There is no BGP peer group.

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
11.0	This command was introduced.
11.1(20)CC	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(2)S	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(7)T	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were removed. Address family configuration mode was added.

Usage Guidelines

Often in a BGP or multiprotocol BGP speaker, many neighbors are configured with the same update policies (that is, same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into peer groups to simplify configuration and make update calculation more efficient.

Note Peer group members can span multiple logical IP subnets, and can transmit, or pass along, routes from one peer group member to another.

Once a peer group is created with the **neighbor peer-group** command, it can be configured with the **neighbor** commands. By default, members of the peer group inherit all the configuration options of the peer group. Members also can be configured to override the options that do not affect outbound updates.

Peer group members will always inherit the following configuration options: remote-as (if configured), version, update-source, out-route-map, out-filter-list, out-dist-list, minimum-advertisement-interval, and next-hop-self. All the peer group members will inherit changes made to the peer group.

If a peer group is not configured with a remote-as, the members can be configured with the **neighbor** {*ip-address* | *peer-group-name*} **remote-as** command. This command allows you to create peer groups containing Exterior Border Gateway Protocol (EBGP) neighbors.

Examples

The following example configurations show how to create these types of neighbor peer group:

- Interior Border Gateway Protocol (IBGP) peer group
- EBGP peer group
- Multiprotocol BGP peer group

IBGP Peer Group

In the following example, the peer group named internal configures the members of the peer group to be IBGP neighbors. By definition, this is an IBGP peer group because the **router bgp** command and the **neighbor remote-as** command indicate the same autonomous system (in this case, autonomous system 100). All the peer group members use loopback 0 as the update source and use set-med as the outbound route map. The **neighbor internal filter-list 2 in** command shows that, except for 171.69.232.55, all the neighbors have filter-list 2 as the inbound filter list.

```
router bgp 100
 neighbor internal peer-group
 neighbor internal remote-as 100
 neighbor internal update-source loopback 0
 neighbor internal route-map set-med out
 neighbor internal filter-list 1 out
 neighbor internal filter-list 2 in
 neighbor 171.69.232.53 peer-group internal
 neighbor 171.69.232.54 peer-group internal
 neighbor 171.69.232.55 peer-group internal
 neighbor 171.69.232.55 filter-list 3 in
```

EBGP Peer Group

The following example defines the peer group named external-peers without the **neighbor remote-as** command. By definition, this is an EBGP peer group because each individual member of the peer group is configured with its respective autonomous system number separately. Thus the peer group consists of members from autonomous systems 200, 300, and 400. All the peer group members have the set-metric route map as an outbound route map and filter-list 99 as an outbound filter list. Except for neighbor 171.69.232.110, all of them have 101 as the inbound filter list.

```
router bgp 100
 neighbor external-peers peer-group
 neighbor external-peers route-map set-metric out
 neighbor external-peers filter-list 99 out
 neighbor external-peers filter-list 101 in
 neighbor 171.69.232.90 remote-as 200
 neighbor 171.69.232.90 peer-group external-peers
 neighbor 171.69.232.100 remote-as 300
 neighbor 171.69.232.100 peer-group external-peers
 neighbor 171.69.232.110 remote-as 400
 neighbor 171.69.232.110 peer-group external-peers
 neighbor 171.69.232.110 filter-list 400 in
```

Multiprotocol BGP Peer Group

In the following example, all members of the peer group are multicast capable:

```
router bgp 100
neighbor 1.1.1.1 remote-as 1
neighbor 2.2.2.2 remote-as 2
address-family ipv4 multicast
 neighbor mygroup peer-group
 neighbor 1.1.1.1 peer-group mygroup
 neighbor 2.2.2.2 peer-group mygroup
 neighbor 1.1.1.1 activate
 neighbor 2.2.2.2 activate
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPNv4 address prefixes.
clear ip bgp peer-group	Removes all the members of a BGP peer group.
show ip bgp peer-group	Displays information about BGP peer groups.

neighbor remote-as

To add an entry to the Border Gateway Protocol (BGP) or multiprotocol BGP neighbor table, use the **neighbor remote-as** command in router configuration mode. To remove an entry from the table, use the **no** form of this command.

neighbor {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*

no neighbor {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*

Syntax Description

ip-address	IP address of the neighbor.
peer-group-name	Name of a BGP peer group.
autonomous-system-number	Autonomous system to which the neighbor belongs.

Defaults

There are no BGP or multiprotocol BGP neighbor peers.

Command Modes

Router configuration

Command History

Release	Modification
10.0	This command was introduced.
11.0	The *peer-group-name* argument was added.
11.1(20)CC	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(7)T	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were removed.

Usage Guidelines

Specifying a neighbor with an autonomous system number that matches the autonomous system number specified in the **router bgp** global configuration command identifies the neighbor as internal to the local autonomous system. Otherwise, the neighbor is considered external.

If you specify a BGP or multiprotocol BGP peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command.

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To exchange other address prefix types, such as multicast and VPNv4, neighbors must also be activated using the **neighbor activate** command in address family configuration mode.

Examples

The following example specifies that a router at the address 131.108.1.2 is a neighbor in autonomous system number 109:

```
router bgp 110
 network 131.108.0.0
 neighbor 131.108.1.2 remote-as 109
```

The following example assigns a BGP router to autonomous system 109, and two networks are listed as originating in the autonomous system. Then the addresses of three remote routers (and their autonomous systems) are listed. The router being configured will share information about networks 131.108.0.0 and 192.31.7.0 with the neighbor routers. The first router listed is in the same Class B network address space, but in a different autonomous system; the second **neighbor remote-as** command illustrates specification of an internal neighbor (with the same autonomous system number) at address 131.108.234.2; and the last **neighbor remote-as** command specifies a neighbor on a different network.

```
router bgp 109
 network 131.108.0.0
 network 192.31.7.0
 neighbor 131.108.200.1  remote-as 167
 neighbor 131.108.234.2  remote-as 109
 neighbor 150.136.64.19  remote-as  99
```

The following example configures neighbor 131.108.1.1 in autonomous system 1 to exchange only multicast routes:

```
router bgp 109
neighbor 131.108.1.1 remote-as 1
neighbor 131.108 1.2 remote-as 1
neighbor 2.2.2.2 remote-as 2
address-family ipv4 multicast
  neighbor 131.108.1.1 activate
  neighbor 131.108.1.2 activate
  neighbor 2.2.2.2 activate
```

The following example configures neighbor 131.108.1.1 in autonomous system 1 to exchange only unicast routes:

```
router bgp 109
neighbor 131.108.1.1 remote-as 1
neighbor 131.108 1.2 remote-as 1
neighbor 2.2.2.2 remote-as 2
```

Related Commands

Command	Description
neighbor peer-group (creating)	Creates a BGP peer group.
router bgp	Configures the BGP routing process.

neighbor route-map

To apply a route map to incoming or outgoing routes, use the **neighbor route-map** command in address family or router configuration mode. To remove a route map, use the **no** form of this command.

neighbor {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}

no neighbor {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}

Syntax Description

ip-address	IP address of the neighbor.
peer-group-name	Name of a BGP or multiprotocol BGP peer group.
route-map-name	Name of route map.
in	Apply to incoming routes.
out	Apply to outgoing routes.

Defaults

No route maps are applied to a peer.

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(7)T	Address family configuration mode was added.

Usage Guidelines

When specified in address family configuration mode, this command applies a route map to that particular address family only. When specified in router configuration mode, this command applies a route map to IPv4 unicast routes only.

If an outbound route map is specified, it is proper behavior to only advertise routes that match at least one section of the route map.

If you specify a BGP or multiprotocol BGP peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command. Specifying the command for a neighbor overrides the inbound policy that is inherited from the peer group.

Examples

The following router configuration mode example applies a route map named internal-map to a BGP incoming route from 198.92.70.24:

```
router bgp 5
 neighbor 198.92.70.24 route-map internal-map in

route-map internal-map
 match as-path 1
 set local-preference 100
```

The following address family configuration mode example applies a route map named internal-map to a multiprotocol BGP incoming route from 198.92.70.24:

```
router bgp 5
address-family ipv4 multicast
 neighbor 198.92.70.24 route-map internal-map in

route-map internal-map
 match as-path 1
 set local-preference 100
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPNv4 address prefixes.
neighbor peer-group (creating)	Creates a BGP peer group.

network (BGP and multiprotocol BGP)

To specify the networks to be advertised by the Border Gateway Protocol (BGP) and multiprotocol BGP routing processes, use the **network** command in address family or router configuration mode. To remove an entry, use the **no** form of this command.

network *network-number* [**mask** *network-mask*]

no network *network-number* [**mask** *network-mask*]

Syntax Description

network-number	Network that BGP or multiprotocol BGP will advertise.
mask	(Optional) Network or subnetwork mask.
network-mask	(Optional) Network mask address.

Defaults

No networks are specified.

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0	The limit of 200 network commands per BGP router was removed.
11.1(20)CC	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were added.
12.0(7)T	The **nlri unicast**, **nlri multicast**, and **nlri unicast multicast** keywords were removed. Address family configuration mode was added.

Usage Guidelines

BGP and multiprotocol BGP networks can be learned from connected routes, dynamic routing, and from static route sources.

The maximum number of **network** commands you can use is determined by the resources of the router, such as the configured NVRAM or RAM.

Examples

The following example sets up network 131.108.0.0 to be included in the BGP updates:

```
router bgp 120
 network 131.108.0.0
```

The following example sets up network 131.108.0.0 to be included in the multiprotocol BGP updates:

```
router bgp 120
address family ipv4 multicast
 network 131.108.0.0
```

Related Commands

Command	Description
address-family ipv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard IPv4 address prefixes.
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPNv4 address prefixes.
default-information originate (BGP)	Allows the redistribution of network 0.0.0.0 into BGP.
neighbor ebgp-multihop	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
network backdoor	Specifies a backdoor route to a BGP border router that will provide better information about the network.
router bgp	Configures the BGP routing process.

redistribute dvmrp

To configure redistribution of Distance Vector Multicast Routing Protocol (DVMRP) routes into multiprotocol Border Gateway Protocol (BGP), use the **redistribute dvmrp** command in address family or router configuration mode. To stop such redistribution, use the **no** form of this command.

redistribute dvmrp [**route-map** *map-name*]

no redistribute dvmrp [**route-map** *map-name*]

Syntax Description

route-map *map-name*	(Optional) Name of the route map that contains various BGP attribute settings.

Defaults

DVMRP routes are not redistributed into multiprotocol BGP.

Command Modes

Address family configuration

Router configuration

Command History

Release	Modification
11.1(20)CC	This command was introduced.
12.0(7)T	Address family configuration mode was added.

Usage Guidelines

Use this command if you have a subset of DVMRP routes in an autonomous system that you want to take the multiprotocol BGP path. Define a route map to further specify which DVMRP routes get redistributed.

Examples

The following router configuration mode example redistributes DVMRP routes to BGP peers that match access list 1:

```
router bgp 109
 redistribute dvmrp route-map dvmrp-into-mbgp
route-map dvmrp-into-mbgp
 match ip address 1
```

The following address family configuration mode example redistributes DVMRP routes to multiprotocol BGP peers that match access list 1:

```
router bgp 109
address-family ipv4 multicast
 redistribute dvmrp route-map dvmrp-into-mbgp

route-map dvmrp-into-mbgp
 match ip address 1
```

set nlri

In Cisco IOS Release 12.0(7)T, the **address-family ipv4** and **address-family vpnv4** commands replace the **set nlri** command. See the descriptions of the **address-family ipv4** or **address-family vpnv4** command for more information.

show ip bgp ipv4 multicast

To display IPv4 multicast database-related information, use the **show ip bgp ipv4 multicast** EXEC command.

show ip bgp ipv4 multicast [*command*]

Syntax Description

command	(Optional) Any multiprotocol Border Gateway Protocol (BGP) command supported by the **show ip bgp ipv4 multicast** command.

Command Modes

EXEC

Command History

Release	Modification
12.0(7)T	This command was introduced.

Usage Guidelines

Use this command in conjunction with the **show ip rpf** command to determine if IP multicast routing is using multiprotocol BGP routes.

To determine which multiprotocol BGP commands are supported by the **show ip bgp ipv4 multicast** command, enter the following while in EXEC mode:

```
Router# show ip bgp ipv4 multicast ?
```

The **show ip bgp ipv4 multicast** command replaces the **show ip mbgp** command.

Examples

The following is sample output from the **show ip bgp ipv4 multicast** command:

```
Router# show ip bgp ipv4 multicast

MBGP table version is 6, local router ID is 192.168.200.66
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.20.16/28     0.0.0.0                  0      0 32768 i
*> 10.0.35.16/28     0.0.0.0                  0      0 32768 i
*> 10.0.36.0/28      0.0.0.0                  0      0 32768 i
*> 10.0.48.16/28     0.0.0.0                  0      0 32768 i
*> 10.2.0.0/16       0.0.0.0                  0      0 32768 i
*> 10.2.1.0/24       0.0.0.0                  0      0 32768 i
*> 10.2.2.0/24       0.0.0.0                  0      0 32768 i
*> 10.2.3.0/24       0.0.0.0                  0      0 32768 i
*> 10.2.7.0/24       0.0.0.0                  0      0 32768 i
*> 10.2.8.0/24       0.0.0.0                  0      0 32768 i
*> 10.2.10.0/24      0.0.0.0                  0      0 32768 i
*> 10.2.11.0/24      0.0.0.0                  0      0 32768 i
*> 10.2.12.0/24      0.0.0.0                  0      0 32768 i
*> 10.2.13.0/24      0.0.0.0                  0      0 32768 i
```

Table 2 describes the significant fields shown in the display.

Table 2 show ip bgp ipv4 multicast Field Descriptions

Field	Description
MBGP table version	Internal version number of the table. This number is incremented whenever the table changes.
local router ID	IP address of the router.
Status codes	Status of the table entry. The status is displayed at the beginning of each line in the table. It can be one of the following values: s—The table entry is suppressed. d—The table entry is dampened. h—The table entry is historical. *—The table entry is valid. >—The table entry is the best entry to use for that network. i—The table entry was learned via an internal BGP session.
Origin codes	Indicates the origin of the entry. The origin code is placed at the end of each line in the table. It can be one of the following values: i—Entry originated from Interior Gateway Protocol (IGP) and was advertised with a **network** router configuration or address family configuration command. e—Entry originated from Exterior Gateway Protocol (EGP). ?—Origin of the path is not clear. Usually, this is a router that is redistributed into BGP from an IGP.
Network	IP address of a network entity.
Next Hop	IP address of the next system that is used when forwarding a packet to the destination network. An entry of 0.0.0.0 indicates that the router has some non-BGP routes to this network.
Metric	If shown, this is the value of the interautonomous system metric.
LocPrf	Local preference value as set with the **set local-preference** route-map configuration command. The default value is 100.
Weight	Weight of the route as set via autonomous system filters.
Path	Autonomous system paths to the destination network. There can be one entry in this field for each autonomous system in the path.

Related Commands

Command	Description
show ip rpf	Displays how IP multicast routing does RPF.

show ip bgp ipv4 multicast summary

To display a summary of IPv4 multicast database-related information, use the **show ip bgp ipv4 multicast summary** EXEC command.

show ip bgp ipv4 multicast summary

Syntax Description

This command has no arguments or keywords.

Command Modes

EXEC

Command History

Release	Modification
12.0(7)T	This command was introduced.

Usage Guidelines

The **show ip bgp ipv4 multicast summary** command replaces the **show ip mbgp summary** command.

Examples

The following is sample output from the **show ip bgp ipv4 multicast summary** command:

```
Router# show ip bgp ipv4 multicast summary

BGP router identifier 10.0.33.34, local AS number 34
BGP table version is 5, main routing table version 1
4 network entries and 6 paths using 604 bytes of memory
5 BGP path attribute entries using 260 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP community entries using 48 bytes of memory
2 BGP route-map cache entries using 32 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 8/28 prefixes, 12/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.33.35      4    35     624     624        5    0    0 10:13:46        3
```

Table 3 describes the significant fields shown in the display.

Table 3 show ip bgp ipv4 multicast summary Field Descriptions

Field	Description
Neighbor	IP address of configured neighbor in the multicast routing table.
V	Version of the multiprotocol Border Gateway Protocol (BGP) used.
AS	Autonomous system to which the neighbor belongs.
MsgRcvd	Number of messages received from the neighbor.
MsgSent	Number of messages sent to the neighbor.
TblVer	Number of the table version, which is incremented each time the table changes.
InQ	Number of messages received in the input queue.

Table 3 show ip bgp ipv4 multicast summary Field Descriptions (continued)

Field	Description
OutQ	Number of messages ready to go in the output queue.
Up/Down	Days and hours that the neighbor has been up or down (no information in the State column means the connection is up).
State/PfxRcd	State of the neighbor/number of routes received. If no state is indicated, the state is up.

Related Commands

Command	Description
show ip rpf	Displays how IP multicast routing does RPF.

show ip mbgp

In Cisco IOS Release 12.0(7)T, the **show ip bgp ipv4 multicast** command replaces the **show ip mbgp** command. See the description of the **show ip bgp ipv4 multicast** command for more information.

show ip mbgp summary

In Cisco IOS Release 12.0(7)T, the **show ip bgp ipv4 multicast summary** command replaces the **show ip mbgp summary** command. See the description of the **show ip bgp ipv4 multicast summary** command for more information.

Debug Commands

This section documents new **debug** commands. All other commands used with multiprotocol Border Gateway Protocol (BGP) are documented in the Cisco IOS Release 12.1 command reference publications.

- **debug ip mbgp dampening**
- **debug ip mbgp updates**

debug ip mbgp dampening

To log route flap dampening activity related to multiprotocol Border Gateway Protocol (BGP), use the **debug ip mbgp dampening** privileged EXEC command. To disable debugging output, use the **no** form of this command.

debug ip mbgp dampening [*access-list-number*]

no debug ip mbgp dampening [*access-list-number*]

Syntax Description

access-list-number	(Optional) Number of an access list in the range from 1 to 99. If an access list number is specified, debugging occurs only for the routes permitted by the access list.

Defaults

Logging for route flap dampening activity is not enabled.

Command History

Release	Modification
11.1(20)CC	This command was introduced.

Examples

The following example shows sample **debug ip mbgp dampening** output:

```
Router# debug ip mbgp dampening

BGP: charge penalty for 173.19.0.0/16 path 49 with halflife-time 15 reuse/suppress
750/2000
BGP: flapped 1 times since 00:00:00. New penalty is 1000
BGP: charge penalty for 173.19.0.0/16 path 19 49 with halflife-time 15 reuse/suppress
750/2000
BGP: flapped 1 times since 00:00:00. New penalty is 1000
```

debug ip mbgp updates

To log multiprotocol Border Gateway Protocol (BGP) -related information passed in BGP Update messages, use the **debug ip mbgp updates** privileged EXEC command. To disable debugging output, use the **no** form of this command.

debug ip mbgp updates

no debug ip mbgp updates

Syntax Description

This command has no arguments or keywords.

Defaults

Logging for multiprotocol BGP-related information in BGP Update messages is not enabled.

Command History

Release	Modification
11.1(20)CC	This command was introduced.

Examples

The following example shows sample **debug ip mbgp updates** output:

```
Router# debug ip mbgp updates

BGP: NEXT_HOP part 1 net 200.10.200.0/24, neigh 171.69.233.49, next 171.69.233.34
BGP: 171.69.233.49 send UPDATE 200.10.200.0/24, next 171.69.233.34, metric 0, path 33 34
19 49 109 65000 297 3561 6503
BGP: NEXT_HOP part 1 net 200.10.202.0/24, neigh 171.69.233.49, next 171.69.233.34
BGP: 171.69.233.49 send UPDATE 200.10.202.0/24, next 171.69.233.34, metric 0, path 33 34
19 49 109 65000 297 1239 1800 3597
BGP: NEXT_HOP part 1 net 200.10.228.0/22, neigh 171.69.233.49, next 171.69.233.34
BGP: 171.69.233.49 rcv UPDATE about 222.2.2.0/24, next hop 171.69.233.49, path 49 109
metric 0
BGP: 171.69.233.49 rcv UPDATE about 131.103.0.0/16, next hop 171.69.233.49, path 49 109
metric 0
BGP: 171.69.233.49 rcv UPDATE about 206.205.242.0/24, next hop 171.69.233.49, path 49 109
metric 0
BGP: 171.69.233.49 rcv UPDATE about 1.0.0.0/8, next hop 171.69.233.49, path 49 19 metric 0
BGP: 171.69.233.49 rcv UPDATE about 198.1.2.0/24, next hop 171.69.233.49, path 49 19
metric 0
BGP: 171.69.233.49 rcv UPDATE about 171.69.0.0/16, next hop 171.69.233.49, path 49 metric
0
BGP: 171.69.233.49 rcv UPDATE about 172.19.0.0/16, next hop 171.69.233.49, path 49 metric
0
BGP: nettable_walker 172.19.0.0/255.255.0.0 calling revise_route
BGP: revise route installing 172.19.0.0/255.255.0.0 -> 171.69.233.49
BGP: 171.69.233.19 computing updates, neighbor version 267099, table version 267100,
starting at 0.0.0.0
BGP: NEXT_HOP part 1 net 172.19.0.0/16, neigh 171.69.233.19, next 171.69.233.49
BGP: 171.69.233.19 send UPDATE 172.19.0.0/16, next 171.69.233.49, metric 0, path 33 49
BGP: 1 updates (average = 46, maximum = 46)
BGP: 171.69.233.19 updates replicated for neighbors : 171.69.233.34, 171.69.233.49,
171.69.233.56
BGP: 171.69.233.19 1 updates enqueued (average=46, maximum=46)
BGP: 171.69.233.19 update run completed, ran for 0ms, neighbor version 267099, start
version 267100, throttled to 267100, check point net 0.0.0.0
```

Glossary

Distance Vector Multicast Routing Protocol—See DVMRP.

DVMRP—Distance Vector Multicast Routing Protocol. Internetwork gateway protocol, largely based on RIP, that implements a typical dense mode IP multicast scheme. DVMRP uses IGMP to exchange routing datagrams with its neighbors.

multiprotocol BGP—multiprotocol Border Gateway Protocol. Multiprotocol BGP is an enhancement of BGP that enables the protocol to specify particular NLRI information for IPv4, IP multicast, and VPNv4. The ability to specify particular NLRI information allows the configuration of different routing policies for each network layer family.

multiprotocol Border Gateway Protocol—See multiprotocol BGP.

network layer reachability information—See NLRI.

NLRI—network layer reachability information. An indication, in the form of an IP prefix route, of the networks being advertised. (The IP prefix is an IP network address with an indication of the number of bits (left to right) that constitute the network number.) The NLRI consists of multiple instances of the 2-tuples (length, prefix) where length is the number of masking bits that a particular prefix has. In BGP4, the NLRI is the mechanism that supports classless routing (see RFC 2283, *Multiprotocol Extensions for BGP-4*).

PIM—Protocol Independent Multicast. Multicast routing architecture that allows the addition of IP multicast routing on existing IP networks. PIM is unicast routing protocol independent and can be operated in two modes: dense and sparse. See also PIM dense mode and PIM sparse mode.

PIM dense mode—See PIM DM.

PIM DM—PIM dense mode. One of the two PIM operational modes. PIM dense mode is data-driven and resembles typical multicast routing protocols. Packets are forwarded on all outgoing interfaces until pruning and truncation occurs. In dense mode, receivers are densely populated, and it is assumed that the downstream networks want to receive and probably will use the datagrams that are forwarded to them. The cost of using dense mode is its default flooding behavior. Sometimes called dense mode PIM. Contrast with PIM sparse mode. See also PIM.

PIM SM—PIM sparse mode. One of the two PIM operational modes. PIM sparse mode tries to constrain data distribution so that a minimal number of routers in the network receive it. Packets are sent only if they are explicitly requested at the rendezvous point (RP). In sparse mode, receivers are widely distributed, and the assumption is that downstream networks will not necessarily use the datagrams that are sent to them. The cost of using sparse mode is its reliance on the periodic refreshing of explicit Join messages and its need for RPs. Sometimes called sparse mode PIM. Contrast with PIM dense mode. See also PIM.

PIM sparse mode—See PIM SM.

Protocol Independent Multicast—See PIM.

Reverse-Path Forwarding—See RPF.

RPF—Reverse Path Forwarding. Multicasting technique in which a multicast datagram is forwarded out of all but the receiving interface if the receiving interface is the one used to forward unicast datagrams to the source of the multicast datagram.