# Multiprotocol BGP

## Feature Summary

Multiprotocol BGP (MBGP) 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, MBGP 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 PIM to build data distribution trees.

It is possible to configure BGP peers that exchange both unicast and multicast network layer reachability information (NLRI).

MBGP 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). MBGP allows you to have a unicast routing topology different from a multicast routing topology. Thus, you have more control over your network and resources.

Prior to MBGP, the only way to do interdomain multicast routing was to use the BGP infrastructure that was in place for unicast routing. If those routers were not multicast capable, or you had differing policies where you wanted multicast traffic to flow, you could not support it.

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

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**

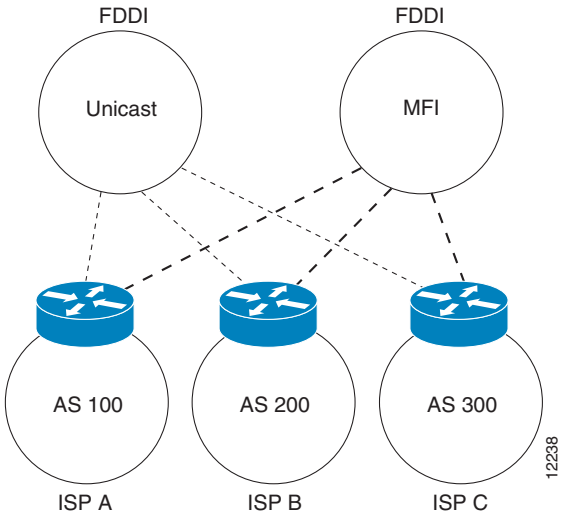


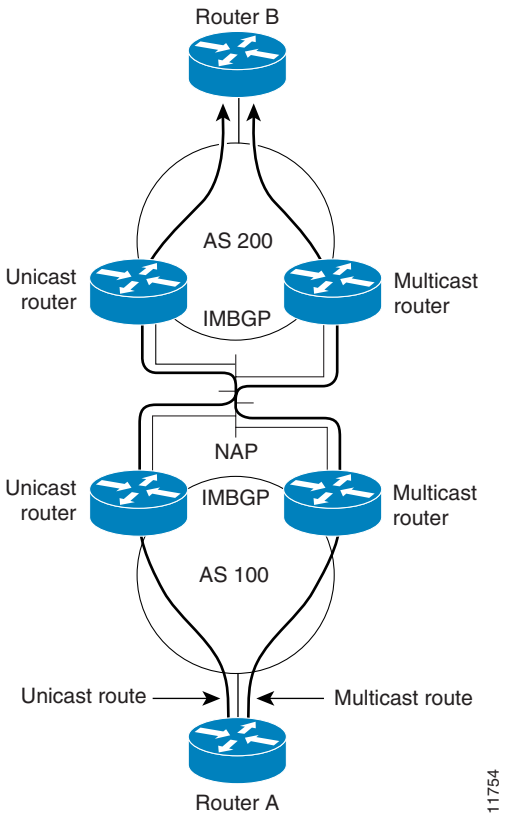

Figure 2 is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they don't support or are not configured to do multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both unicast and multicast routing. The four routers in the middle 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 an MBGP environment with a separate unicast route and multicast route from Router A to Router B. MBGP allows these routes to be non-congruent. Both of the autonomous systems must be configured for Internal MBGP (IMBGP).

A multicast routing protocol, such as PIM, uses the MBGP routing table 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 Multiprotocol BGP Environment**




# Benefits

MBGP offers the following benefits:

- An internet can support incongruent unicast and multicast topologies.
- An internet can support congruent unicast and multicast topologies that have different policies (BGP filtering configurations).
- All of the routing policy capabilities of BGP can be applied to MBGP.
- All of the BGP commands can be used with MBGP.

# Restriction

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

# Platforms

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

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

# 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 Release 11.3 *Network Protocols Configuration Guide, Part 1*.

# Supported MIBs and RFCs

In the future, the BGP MIB will be updated to support reading the MBGP routing table.

This feature supports RFC 2283, Multiprotocol Extensions for BGP-4.

# Configuration Tasks

Perform the following tasks to configure MBGP. The first task is required; the remaining tasks are optional.

- Enable an MBGP Peer
- Enable MBGP Peer Groups
- Advertise Routes into MBGP
- Configure Aggregate MBGP Addresses
- Create a Route Map Based on Unicast or Multicast NLRI
- Change MBGP Administrative Distances
- Configure DVMRP Interoperability with MBGP
- Monitor MBGP

## Enable an MBGP Peer

To enable MBGP between two routers, you must perform these tasks at a minimum on each router, beginning in global configuration mode:

| Task | | Command |
|---|---|---|
| **Step 1** | Configure BGP routing. | **router bgp** *autonomous-system* |
| **Step 2** | Identify the neighbor in the remote AS to the local router, and configure the path between them as MBGP. | **neighbor** {*ip-address* \| *peer-group-name*} **remote-as** *number* **nlri multicast** |

# Enable MBGP Peer Groups

To enable a peer group to perform MBGP routing, perform the following tasks in router configuration mode:

| | | |
|---|---|---|
| **Step 1** | Enable the specified peer group to perform MBGP routing. | **neighbor** *peer-group-name* **peer-group nlri multicast** |
| **Step 2** | Assign options to the peer group. | Refer to the "Configure BGP Peer Groups" section of the "Configuring BGP" chapter in the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |
| **Step 3** | Make a BGP neighbor a member of the peer group. | Refer to the "Configure BGP Peer Groups" section of the "Configuring BGP" chapter of the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |

# Advertise Routes into MBGP

You would advertise routes into MBGP when there are multicast sources in an MBGP routing domain that want their packets to be received by receivers outside their domain.

There are two ways to advertise routes from the local AS into the MBGP routing table.

- Explicitly configure which routes to advertise (use the **network** command in the following table).
- Redistribute routes by protocol (use the **redistribute** command).

We recommend using the **network** command rather than the **redistribute** command to inject routes into MBGP. Perform either of the following tasks in router configuration mode:

| Task | Command |
|---|---|
| Advertise (inject) this network number and mask into the multicast Routing Information Base (RIB). (The routes must first be found in the unicast forwarding table). Tag routes from the specified network as "local origin." | **network** *network-number* [**mask** *mask*] **nlri multicast** |
| Redistribute routes from a specified network into the MBGP routing table. Tag the routes as "incomplete origin." | **redistribute** *protocol* [*process-id*] {**level-1 \| level-1-2 \| level-2**}[**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal \| external 1 \| external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**] |

To redistribute DVMRP routes into MBGP, see the section "Redistribute DVMRP Routes into MBGP" later in this document.

# Configure Aggregate MBGP Addresses

You can configure aggregate MBGP addresses the same way you configure aggregate addresses for unicast BGP. The **aggregate-address** command is enhanced to specify whether the aggregate address should be applied to the multicast RIB, the unicast RIB, or both. If no NLRI type is specified, by default the aggregate address is applied to the unicast RIB only.

To configure an aggregate address for MBGP, perform the following task in router configuration mode:

| Task | Command |
|---|---|
| Configure an aggregate address with various options, including the NLRI type (which determines whether the aggregate address applies to unicast, MBGP, or both). | **aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**nlri** {**unicast** \| **multicast** \| **unicast multicast**}] |

# Create a Route Map Based on Unicast or Multicast NLRI

You can use route map **match** or **set** conditions in conjunction with NLRI unicast or multicast designations. There are two ways to use route maps in conjunction with MBGP:

- Matching on NLRI Type—Create a route-map with the condition that if the type of Routing Information Base (RIB) (unicast, multicast, or both) matches, then subsequent **set** statements will be performed.
- Setting the NLRI Type—Create a route-map such that if the match conditions occur, the route will be injected into the unicast or multicast RIB, or both.

Each of these methods is explained in the following sections.

**Note** It does not make sense to use the **match nlri** command and the **set nlri** command in the same route-map. Use the **match nlri** command with other **set** commands, or use the **set nlri** command with other **match** commands.

## Matching on NLRI Type

Using a route map, you can select specific routes from a BGP Update message by specifying a **match nlri** clause (either unicast, multicast, or both). Then proceed to set the characteristics you want.

Another reason to match on NLRI type is if you are filtering routing information you send to a neighbor who is configured as both a unicast and multicast peer. You can select from which RIB you do the filtering. This is done with the **neighbor route-map out** command.

Specifying a route map containing a **match nlri multicast** command means only multicast table entries will be subject to the subsequent **set** commands. Likewise, a **match nlri unicast** command will match only unicast table entries. If no **match nlri** command is specified, the default is **nlri unicast multicast**, meaning that either unicast or multicast routes will match.

To create a route map and match on NLRI type, perform the following tasks beginning in global configuration mode:

| Task | | Command |
|---|---|---|
| **Step 1** | Define a route map. | **route-map** *map-tag* [**permit** \| **deny**] [*sequence-number*] |
| **Step 2** | In route-map configuration mode, match routes based on whether they are unicast, multicast, or either one. | **match nlri** {**unicast** \| **multicast** \| **unicast multicast**} |
| **Step 3** | Optionally, continue to define the route map with additional match criteria. | For various **match** commands, refer to the "Configuring BGP" chapter or the "Configuring IP Routing Protocol-Independent Features" chapter in the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |

| Task | | Command |
|---|---|---|
| **Step 4** | Specify any **set** commands (BGP-specific or protocol-independent). | For various **set** commands, refer to the "Configuring BGP" chapter or the "Configuring IP Routing Protocol-Independent Features" chapter in the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |

You can use the **match nlri** command in conjunction with the **neighbor route-map in | out** command so one route-map reference describes filtering policies for different NLRI types (unicast or multicast). That is, the **match nlri** designation determines your outgoing NLRI policy (what type of routes you send the neighbor).

## Setting the NLRI Type

You can elect to inject routes (that meet the match criteria) specifically into the unicast RIB, multicast RIB, or both, by setting the NRLI type.

Specifying a route map that contains any **match** commands and then the **set nlri** command means any routes that pass the match criteria will be injected into the unicast or multicast RIB, as specified by the keywords **unicast**, **multicast**, or both keywords. If both keywords are specified, the route is injected into both RIBs and advertised as separate NLRI in a BGP Update message. If no **set nlri** command is specified, the default is **nlri unicast**.

Use the **set nlri** command in a route map that is referenced by any of these router configuration commands: **aggregate-address**, **neighbor default-originate route-map**, or **redistribute**.

If you want to configure the sending of a default route to a neighbor, use the **set nlri** command in conjunction with the **neighbor default-originate route-map** command. If the **set nlri** is supplied in the route-map referenced by the **neighbor default-originate route-map** command, the multicast default route can be generated independent of the unicast default route.

To create a route map and specify that any matches are injected into either the unicast RIB, multicast RIB, or both, perform the following tasks beginning in global configuration mode:

| Task | | Command |
|---|---|---|
| **Step 1** | Define a route map. | **route-map** *map-tag* [**permit | deny**] [*sequence-number*] |
| **Step 2** | In route-map configuration mode, specify any **match** commands (BGP-specific or protocol-independent). | For various **match** commands, refer to the "Configuring BGP" chapter or the "Configuring IP Routing Protocol-Independent Features" chapter in the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |
| **Step 3** | Inject the routes that meet the match criteria into the unicast RIB, the multicast RIB, or both. | **set nlri** {**unicast | multicast | unicast multicast**} |
| **Step 4** | Optionally, continue with additional **set** commands. | For various **set** commands, refer to the "Configuring BGP" chapter or the "Configuring IP Routing Protocol-Independent Features" chapter in the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*. |

# Change MBGP Administrative Distances

Administrative distance is a way to prefer one routing table over another if the same prefix appears in multiple tables. The lower the value, the more preferred the route is. By default, routes learned from external MBGP are most preferred, with a value of 20. Routes learned from internal MBGP are set to 200; routes that are part of the local autonomous system also default to 200.

IP multicast routing needs to know which route to use if a multicast source is found in any of the following:

- A unicast routing table (which includes tables for IP routing protocols, including BGP external, internal, and local routes)
- An MBGP routing table (which includes external, internal, and local routes)
- A DVMRP routing table
- A static mroute table

The administrative distance allows you to control which routing table you prefer, so keep all of these tables in mind. You can change the administrative distance of MBGP routes, thus making the MBGP routing table preferred over any other table the router is maintaining. If you are running a router with both unicast BGP and MBGP, you should set distances to prefer MBGP.

To change the administrative distances, perform the following task in router configuration mode. Changing the administrative distance of MBGP internal routes is not recommended. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.

| | |
|---|---|
| Change any of the MBGP administrative distances (external distance, internal distance, local distance). | **distance mbgp** *external-distance internal-distance local-distance* |

# Configure DVMRP Interoperability with MBGP

Cisco multicast routers using PIM can interoperate with non-Cisco multicast routers that use 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 who want their packets to be received by receivers outside of this routing domain. MBGP allows the source prefixes of those sources to be known outside of the routing domain.

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

## Redistribute MBGP Routes into DVMRP

By default, no MBGP routes are redistributed to DVMRP. However, you can configure all MBGP routes to be redistributed to DVMRP with a specified metric. Furthermore, to subject only certain MBGP 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 MBGP routes. If the route passes the route map, then the route is redistributed to DVMRP.

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

Perform the following task in interface configuration mode:

| Task | Command |
|---|---|
| Redistribute MBGP routes into DVMRP with a specified metric. An optional route map controls which routes are redistributed; otherwise, all MBGP routes are redistributed. | **ip dvmrp metric** *metric* [**route-map** *map-name*] **mbgp** |

## Redistribute DVMRP Routes into MBGP

If there are multicast sources in a DVMRP routing domain that need to reach receivers in MBGP routing domains, you need to redistribute DVMRP prefixes into MBGP.

If you supply a route map, you can also use the **set** commands to specify various BGP attribute settings. Perform this task in router configuration mode:

| Task | Command |
|---|---|
| Redistribute DVMRP routes into MBGP. | **redistribute dvmrp** [**route-map** *map-name*] |

# Monitor MBGP

Perform any of the following tasks in EXEC mode to monitor information related to MBGP:

| Task | Command |
|---|---|
| Display information related to the multicast RIB, depending on which BGP command you specify. | **show ip mbgp** *command* |
| Display a summary of multicast RIB information. | **show ip mbgp summary** |
| Log route flap dampening activity. | **debug ip mbgp dampening** [*access-list-number*] |
| Log MBGP-related information passed in BGP Update messages. | **debug ip mbgp updates** |

Perform the following task in EXEC mode to monitor RTP information:

| Task | Command |
|---|---|
| Display the quality of RTP data based on packets captured in the IP multicast cache header buffer. | **show ip mpacket** *group-address* [*source-address*] **quality** |

# Configuration Examples

This section provides the following MBGP configuration examples:

- Configure an MBGP Peer Group Example
- Match on NLRI Type Example
- Redistribute DVMRP Routes

## Configure an MBGP Peer Group Example

In the following example, all members of the peer group are both unicast and multicast NLRI-capable:

```
router bgp 100
 neighbor mygroup peer-group nlri unicast multicast
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 peer-group mygroup
 neighbor 2.2.2.2 remote-as 2
 neighbor 2.2.2.2 peer-group mygroup
```

## Match on NLRI Type Example

In the following example, any *multicast* routes (because of the **match nlri multicast** command) from neighbor 1.1.1.1 will be accepted if they match access list 1:

```
router bgp 109
 neighbor 1.1.1.1 remote-as 1 nlri unicast multicast
 neighbor 1.1.1.1 route-map in filter-some-multicast
route-map filter-some-multicast
 match nlri multicast
 match ip address 1
```

## Redistribute DVMRP Routes

The following 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
 set nlri multicast
```

# Command Reference

This section documents the following new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 11.3 command references.

- **aggregate-address**
- **distance mbgp**
- **ip dvmrp metric**
- **ip multicast cache-headers**
- **match nlri**
- **neighbor peer-group (creating)**
- **neighbor remote-as**
- **network (BGP)**
- **redistribute dvmrp**
- **set nlri**
- **show ip mbgp**

- **show ip mbgp summary**
- **show ip mpacket quality**

# aggregate-address

To create an aggregate entry in a BGP or MBGP routing table, use the **aggregate-address** router configuration command. To disable this feature, 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*] [**nlri** {**unicast | multicast | unicast multicast**}]
**no aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**nlri** {**unicast | multicast | unicast multicast**}]

## 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 route map used to select the routes to be suppressed. |
| **advertise-map** *map-name* | (Optional) Name of 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. |
| **nlri unicast** | (Optional) Applies the aggregate to the unicast RIB only. |
| **nlri multicast** | (Optional) Applies the aggregate to the multicast RIB only. |
| **nlri unicast multicast** | (Optional) Applies the aggregate to the unicast and multicast RIBs. |

## Default

Disabled

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0. The **nlri** keywords first appeared in Cisco IOS Release 11.1(20)CC.

You can implement aggregate routing in BGP either by redistributing an aggregate route into 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 routing table if there are any more-specific BGP routes available that fall in the specified range. The aggregate route will be advertised as coming from your autonomous system and has 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 **aggregate-address** when aggregating many paths, because this route must be continually withdrawn and re-updated 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 will also suppress 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 speakers 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, an aggregate address is created. The path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized.

```
router bgp 5
 aggregate-address 193.0.0.0 255.0.0.0 as-set
```

In the following MBGP example, an aggregate address is created and applied to the multicast RIB only. More-specific routes are filtered from updates.

```
router bgp 5
 aggregate-address 193.0.0.0 255.0.0.0 summary-only nlri multicast
```

## Related Commands

**match as-path**
**match ip address**
**route-map**

# distance mbgp

To configure administrative distance for MBGP routes, use the **distance mbgp** router configuration command. To restore the default values, use the **no** form of this command.

**distance mbgp** *external-distance internal-distance local-distance*
**no distance mbgp** *external-distance internal-distance local-distance*

## Syntax Description

| | |
|---|---|
| *external-distance* | Administrative distance for MBGP 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 0 to 255. The default is 20. Routes with a distance of 255 are not installed in the routing table. |
| *internal-distance* | Administrative distance for MBGP internal routes. Internal routes are those routes that are learned from another MBGP entity within the same autonomous system. Acceptable values are from 0 to 255. The default is 200. Routes with a distance of 255 are not installed in the routing table. |
| *local-distance* | Administrative distance for MBGP local routes. Local routes are those networks listed with a **network** router configuration command, often as back doors, for that route or for networks that are being redistributed from another process. Acceptable values are from 0 to 255. The default is 200. Routes with a distance of 255 are not installed in the routing table. |

## Default

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

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

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 an integer between 0 and 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 when one router is running both unicast BGP and MBGP. 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 MBGP, or if some internal routes should really be preferred by MBGP.

**Caution** Changing the administrative distance of MBGP internal routes should be done with caution. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.

The three distance arguments in the command syntax are position dependent; therefore, you must specify a value for all three, even if two of them are the default values.

## Example

The following example configures the router to prefer MBGP routes (external, internal, and local) over BGP routes.

```
router bgp 109
 distance bgp 200 200 200
 distance mbgp 0 0 0
```

## Related Commands

**distance bgp**

# ip dvmrp metric

To configure the metric associated with a set of destinations for DVMRP reports, use one of the **ip dvmrp metric** interface configuration commands. Note that this command has two different syntax possibilities. To disable this function, use the **no** form of this command.

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

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

## 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). |
| **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**.<br>If you specify these arguments, 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. |
| **route-map** *map-name* | 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 multiprotocol BGP (MBGP) routes into DVMRP. |

## Default

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

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.2. The **route-map** keyword first appeared in Cisco IOS Release 11.1. The **mbgp** keyword first appeared in Cisco IOS Release 11.1(20)CC.

When 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 MBGP routes to DVMRP neighbors with a metric of 1:

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

## Related Commands

**debug ip dvmrp**
**ip dvmrp accept-filter**

# ip multicast cache-headers

To allocate a circular buffer to store IP 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) Cache RTP headers. |

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1. The **rtp** keyword was added in Release 11.1(20)CC.

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

- Who is sending IP multicast packets to what groups
- Inter-packet delay
- Duplicate IP multicast packets (if any)
- Multicast forwarding loops in your network (if any)
- Scope of the group
- UDP port numbers
- Packet length

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

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

## Example

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

```
ip multicast cache-headers
```

## Related Commands

**show ip mpacket**
**show ip mpacket quality**

# match nlri

To match a unicast or multicast Routing Information Base (RIB) entry, use the **match nlri** route-map configuration command. To remove a path list entry, use the **no** form of this command.

**match nlri** {**unicast | multicast | unicast multicast**}
**no match nlri** {**unicast | multicast | unicast multicast**}

## Syntax Description

| | |
|---|---|
| **unicast** | Matches a unicast NLRI from incoming Update messages or RIB for outgoing filters being processed for a route map. |
| **multicast** | Matches a multicast NLRI from incoming Update messages or RIB for outgoing filters being processed for a route map. |
| **unicast multicast** | Matches either a unicast or multicast NLRI from incoming Update messages or RIB for outgoing filters being processed for a route map. This is the default setting. |

## Default

The default is **unicast multicast**, meaning that either a unicast or multicast RIB entry will match.

## Command Mode

Route-map configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

This command can be used in conjunction with the **neighbor route-map in** command so you can use one route-map reference to describe filtering policies for different NLRI types.

After stating the match criteria, use any of the set commands in the "Configuring BGP" chapter or the "Configuring IP Routing Protocol-Independent Features" chapter of the Cisco IOS Release 11.3 *Network Protocols Configuration Guide, Part 1*.

## Example

In the following example, any *multicast* routes (because of the **match nlri multicast** command) from neighbor 1.1.1.1 will be accepted if they match access list 1:

```
router bgp 109
 neighbor 1.1.1.1 remote-as 1 nlri unicast multicast
 neighbor 1.1.1.1 route-map in filter-some-multicast
route-map filter-some-multicast
 match nlri multicast
 match ip address 1
```

## Related Commands

**route-map**

# neighbor peer-group (creating)

To create a BGP peer group, use the **neighbor peer-group** router configuration command. To remove the peer group and all of its members, use the **no** form of this command.

**neighbor** *peer-group-name* **peer-group** [**nlri** {**unicast | multicast | unicast multicast**}]
**no neighbor** *peer-group-name* **peer-group** [**nlri** {**unicast | multicast | unicast multicast**}]

## Syntax Description

| | |
|---|---|
| *peer-group-name* | Name of the BGP peer group. |
| **nlri unicast** | (Optional) Only unicast network layer reachability information (NLRI) will be sent to the neighbor. This is the default value. |
| **nlri multicast** | (Optional) Only multicast NLRI will be sent to the neighbor. |
| **nlri unicast multicast** | (Optional) Both unicast and multicast NLRI will be sent to the neighbor. |

## Default

There is no BGP peer group. If a peer group is identified, but no NLRI designation is specified, the default is **nlri unicast**.

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.0. The **nlri** {**unicast | multicast | unicast multicast**} keywords first appeared in Release 11.1(20)CC.

Often in a BGP speaker, there are many neighbors 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.

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 can also 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 EBGP neighbors.

Use the **nlri** keyword with either **unicast**, **multicast**, or **unicast multicast** to specify what type of traffic the peer group can support.

### Example for an 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, AS 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
```

### Example for an EBGP Peer Group

In the following example, the peer group *external-peers* is defined without the **neighbor remote-as** command. This is what makes it an EBGP peer group. Each individual member of the peer group is configured with its respective AS-number separately. Thus the peer group consists of members from autonomous systems 200, 300, and 400. All the peer group members have *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
```

### Example for an MBGP Peer Group

In the following example, all members of the peer group are both unicast and multicast NLRI-capable.

```
router bgp 100
 neighbor mygroup peer-group nlri unicast multicast
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 peer-group mygroup
 neighbor 2.2.2.2 remote-as 2
 neighbor 2.2.2.2 peer-group mygroup
```

### Related Commands

**clear ip bgp peer-group**
**neighbor peer-group (assigning members)**
**show ip bgp peer-group**

# neighbor remote-as

To add an entry to the BGP neighbor table, use the **neighbor remote-as** router configuration command. To remove an entry from the table, use the **no** form of this command.

**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* [**nlri** {**unicast** | **multicast** | **unicast multicast**}]
**no neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* [**nlri** {**unicast** | **multicast** | **unicast multicast**}]

## Syntax Description

| | |
|---|---|
| *ip-address* | Neighbor's IP address. |
| *peer-group-name* | Name of a BGP peer group. |
| *number* | Autonomous system to which the neighbor belongs. |
| **nlri unicast** | (Optional) Only unicast network layer reachability information (NLRI) will be sent to the neighbor. Unicast NLRI is sent in the conventional BGP encoding. If no NLRI designation is specified, **nlri unicast** is the default value. |
| **nlri multicast** | (Optional) Only multicast NLRI will be sent to the neighbor. |
| **nlri unicast multicast** | (Optional) Both unicast and multicast NLRI will be sent to the neighbor. |

## Default

There are no BGP neighbor peers. If a neighbor is identified, but no NLRI designation is specified, the default is **nlri unicast**.

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0. The *peer-group-name* argument first appeared in Cisco IOS Release 11.0. The **nlri** {**unicast** | **multicast** | **unicast multicast**} keywords first appeared in Release 11.1(20)CC.

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 peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command.

If multiprotocol BGP is configured, multicast NLRI is sent in the MP_REACH and MP_UNREACH path attributes. MBP negotiates NLRI in the Capabilities Option of the Open message. Both sides of the BGP connection don't have to be configured consistently. If one side is unicast and multicast, and the other side is multicast only, the session will come up in multicast-only mode.

The only time the connection won't come up is when one side is multicast-only and the other is unicast-only.

## 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
```

In the following example, a BGP router is assigned 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** command illustrates specification of an internal neighbor (with the same autonomous system number) at address 131.108.234.2; and the last **neighbor** 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 AS 1 to exchange only multicast routes. It configures neighbor 131.108.1.2 in AS 1 to exchange both multicast and unicast routes.

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

## Related Commands

**neighbor peer-group (creating)**

# network (BGP)

To specify the list of networks for the BGP routing process, use this form of the **network** router configuration command. To remove an entry, use the **no** form of this command.

**network** *network-number* [**mask** *network-mask*] [**nlri** {**unicast | multicast | unicast multicast**}]
**no network** *network-number* [**mask** *network-mask*] [**nlri** {**unicast | multicast | unicast multicast**}]

## Syntax Description

| | |
|---|---|
| *network-number* | Network that BGP will advertise. |
| **mask** | Network or subnetwork mask. |
| *network-mask* | (Optional) Network mask address. |
| **nlri unicast** | (Optional) The specified network is injected into the unicast RIB only. NLRI unicast is the default value if no NLRI designation is specified. |
| **nlri multicast** | (Optional) The specified network is injected into the multicast RIB only. |
| **nlri unicast multicast** | (Optional) The specified network is injected into both the unicast and multicast RIBs. |

## Default

No networks are specified.

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0. The **nlri** {**unicast | multicast | unicast multicast**} keywords first appeared in Release 11.1(20)CC.

These types of networks can be learned from connected routes, dynamic routing, and from static route sources.

A maximum of 200 **network** commands may be specified for a single BGP process.

## 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 advertises neighbor 131.108.0.0 with the mask to its multicast peers with a local origin attribute. It advertises neighbor 131.140.0.0 with the mask to multicast and unicast peers with a local origin attribute.

```
router bgp 120
 network 131.108.0.0 mask 255.255.0.0 nlri multicast
 network 131.140.0.0 mask 255.255.0.0 nlri multicast unicast
 neighbor 131.150 0.0 mask 255.255.0.0
```

## Related Commands

**network backdoor**
**network mask**
**network weight**
**router bgp**

# redistribute dvmrp

To configure redistribution of DVMRP routes into MBGP, use the **redistribute dvmrp** router configuration command. 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. |

## Default

DVMRP routes are not redistributed into MBGP.

## Command Mode

Router configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

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

## Example

The following 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
 set nlri multicast
```

# set nlri

To inject a route into the unicast or multicast Routing Information Base (RIB), use the **set nlri** route-map configuration command. To remove the set instruction, use the **no** form of this command.

**set nlri** {**unicast** | **multicast**}
**no set nlri** {**unicast** | **multicast**}

## Syntax Description

| | |
|---|---|
| **unicast** | Injects a route that passes the match criteria into the unicast RIB. |
| **multicast** | Injects a route that passes the match criteria into the multicast RIB. |

## Default

**unicast**

## Command Mode

Route-map configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

If both unicast and multicast are specified, the route is injected into both RIBs and advertised as separate NLRI in a BGP Update message. This command is used when referencing a route-map by various router configuration commands, such as **redistribute**, **aggregate-address**, and **neighbor route-map out** references.

This command can be used in conjunction with the **neighbor default-originate route-map** command. If the **set nlri** is supplied in the **route-map** referenced by the **neighbor** command, the multicast default route can be generated independent of the unicast default route.

## Example

The following example redistributes OSPF routes to MBGP peers (because of the **set nlri multicast** command) that match access list 1:

```
router bgp 109
 redistribute ospf 109 route-map ospf-into-mbgp
route-map ospf-into-mbgp
 match ip address 1
 set nlri multicast
```

## Related Commands

**route-map**

# show ip mbgp

To display multicast Routing Information Base (RIB) related information, use the **show ip mbgp** EXEC command.

**show ip mbgp** [*command*]

## Syntax Description

| | |
|---|---|
| *command* | (Optional) Any BGP command supported by **show ip bgp** *command*. |

## Command Mode

EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20) CC.

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

## Sample Display

The following is sample output from the **show ip mbgp** command:

```
Router# show ip mbgp

MBGP table version is 605299, local router ID is 198.9.200.66
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
*> 4.0.20.16/28     0.0.0.0                0      0      0 i
*> 4.0.35.16/28     0.0.0.0                0      0      0 i
*> 4.0.36.0/28      0.0.0.0                0      0      0 i
*> 4.0.48.16/28     0.0.0.0                0      0      0 i
*> 9.2.0.0/16       0.0.0.0                0      0      0 i
*> 9.2.1.0/24       0.0.0.0                0      0      0 i
*> 9.2.2.0/24       0.0.0.0                0      0      0 i
*> 9.2.3.0/24       0.0.0.0                0      0      0 i
*> 9.2.7.0/24       0.0.0.0                0      0      0 i
*> 9.2.8.0/24       0.0.0.0                0      0      0 i
*> 9.2.10.0/24      0.0.0.0                0      0      0 i
*> 9.2.11.0/24      0.0.0.0                0      0      0 i
*> 9.2.12.0/24      0.0.0.0                0      0      0 i
*> 9.2.13.0/24      0.0.0.0                0      0      0 i
```

Table 1 describes the significant fields shown in the display.

**Table 1 Show IP MBGP 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. |

**Table 1 Show IP MBGP Field Descriptions (Continued)**

| Field | Description |
|---|---|
| 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:<br>s—The table entry is suppressed.<br>d—The table entry is dampened.<br>h—The table entry is historical.<br>*—The table entry is valid.<br>>—The table entry is the best entry to use for that network.<br>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:<br>i—Entry originated from IGP and was advertised with a **network** router configuration command.<br>e—Entry originated from EGP.<br>?—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. This field is frequently not used. |
| 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

**show ip rpf**

# show ip mbgp summary

To display a summary of multicast Routing Information Base (RIB) related information, use the **show ip mbgp summary** EXEC command.

**show ip mbgp summary**

## Syntax Description

This command has no arguments or keywords.

## Command Mode

EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

## Sample Display

The following is sample output from the **show ip mbgp summary** command.

```
Router# show ip mbgp summary

MBGP table version is 605471
6521 network entries (6522/19563 paths) using 1458596 bytes of memory
9 BGP path attribute entries using 1048 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
0 prefixes revised.

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
198.9.200.65    4 10888   44088  110375   605471    0    0 3d00h           7
198.9.201.2     4    24   41919  119758   605471    0    0 4d07h           8
```

Table 2 describes the fields in the display.

**Table 2 Show IP MBGP Summary Field Descriptions**

| Field | Description |
|---|---|
| Neighbor | IP address of configured neighbor in the multicast routing table. |
| V | Version of 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. |
| 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. |

# show ip mpacket quality

To display Real-time Transport Protocol (RTP) data quality based on packets captured in the IP multicast cache header buffer, use the **show ip mpacket quality** EXEC command.

**show ip mpacket** *group-address* [*source-address*] **quality**

## Syntax Description

| | |
|---|---|
| *group-address* | Address of the multicast group for which to display information. |
| *source-address* | (Optional) Source address for which to display information. |

## Command Mode

EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(20)CC.

Use this command to monitor loss characteristics of a multimedia stream. Note the following restrictions:

- This command is available only if the **ip multicast cache-headers rtp** command is configured to capture RTP headers.
- Only non-encrypted multimedia streams can be monitored.

## Sample Display

The following is sample output from the **show ip mpacket quality** command:

```
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
```

Table 3 describes the significant fields in the display.

**Table 3 Show IP Mpacket Quality Field Descriptions**

| Field | Description |
|---|---|
| Session | Name of session. |
| Source | IP address and name of the source sending to the multicast group. |
| Port | Port number the information is being received on. |
| Packets received | Number of packets received. |
| lost | Number of packets lost. |
| loss percentage | Number lost divided by total number of packets received and lost. |

**Table 3 Show IP Mpacket Quality Field Descriptions**

| Field | Description |
|---|---|
| Packets misordered | Packets received out of order. |
| average loss gap | Number of RTP packets lost between two received packets. For example, if the router receives packets with the following sequence numbers: 1, 2, 3, 5, 6, 7, 10, 11, 12, 14, 15, there is a gap of 2 (after packet 3), 3 (after packet 7), and 2 (after packet 12). So the average loss gap is (2+3+2)/3 = 2.33. |

## Related Commands

**ip multicast cache-headers**

# Debug Commands

This section documents the following new **debug** commands associated with MBGP:

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

# debug ip mbgp dampening

To log route flap dampening activity related to MBGP, use the **debug ip mbgp dampening** EXEC command.

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

## Syntax Description

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

## Sample Display

Figure 3 below shows sample **debug ip mbgp dampening** output.

**Figure 3 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 MBGP-related information passed in BGP Update messages, use the **debug ip mbgp updates** EXEC command.

[**no**] **debug ip mbgp updates**

## Sample Display

Figure 4 below shows sample **debug ip mbgp updates** output.

**Figure 4 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
```