

Configuring IP Multicasting

3

IP multicasting allows a device to send packets to a group of hosts rather than to a list of individual hosts. This chapter describes how to configure IP multicasting on the ERX system.

Topic	Page
Overview	3-2
References	3-3
Before You Begin	3-3
Enabling IP Multicasting	3-3
Deleting Multicast Forwarding Entries	3-4
Monitoring IP Multicast Settings	3-4
Reverse Path Forwarding	3-9
Multicast Packet Forwarding	3-10
IGMP	3-12
IGMP Proxy	3-24
PIM	3-30
DVMRP	3-52
BGP Multicasting	3-67
Investigating Multicast Routes	3-68

Overview

IPv4 defines three types of addresses: *unicast*, *broadcast*, and *multicast*. Each type of address enables a device to send datagrams to selected recipients:

- A unicast address enables a device to send a datagram to a single recipient.
- A broadcast address enables a device to send a datagram to all hosts on a subnetwork.
- A multicast address enables a device to send a datagram to a specified set of hosts, known as a multicast group, in different subnetworks.

Multicast IP packets contain a Class D address in the Destination Address fields of their headers. A Class D address is the IP address of a multicast group. Refer to *Chapter 2, Configuring IP*, and to *IGMP*, later in this chapter, for information about Class D addresses.

IP multicasting improves network efficiency by allowing a host to transmit a datagram to a targeted group of receivers. For example, a host may want to send a large video clip to a group of selected recipients. It would be time-consuming for the host to unicast the datagram to each recipient individually. If the host broadcasts the video clip throughout the network, network resources are not available for other tasks. The host uses only the resources it needs by multicasting the datagram.

Routers use multicast routing algorithms to determine the best route and transmit multicast datagrams throughout the network. The ERX system supports a number of IP multicasting protocols on virtual routers (VRs). Each VR handles the interoperability of IP multicasting protocols automatically. To start multicast operation on a VR, you access the context for that VR, and configure the desired protocols on the selected interfaces. Table 3-1 lists the protocols the system supports and the function of each protocol.

Table 3-1 Function of multicast protocols on a router

Protocol	Function
Internet Group Membership Protocol (IGMP)	Discovers hosts that belong to multicast group.
Protocol Independent Multicast Protocol (PIM)	Discovers other multicast routers that should receive multicast packets.
Distance Vector Multicast Routing Protocol (DVMRP)	Routes multicast datagrams within autonomous systems.

Table 3-1 Function of multicast protocols on a router (continued)

Protocol	Function
BGP Multicasting Protocol	Routes multicast datagrams between autonomous systems.

The system supports up to 16,384 multicast forwarding entries (multicast routes) at any time.

References

- A “traceroute” Facility for IP Multicast – draft-ietf-idmr-traceroute-ipm-07.txt (January 2001 expiration)
- Distance Vector Multicast Routing Protocol – draft-ietf-idmr-dvmrp-v3-10 (February 2001 expiration)
- IGMP-based Multicast Forwarding ("IGMP Proxying") – draft-ietf-magma-igmp-proxy-00.txt (May 2002 expiration)
- Protocol Independent Multicast MIB for IPv4 – draft-ietf-idmr-pim-mib-10.txt (July 2000 expiration)
- RFC 2362 – Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (June 1998)
- RFC 2236 – Internet Group Management Protocol, Version 2 (November 1997)
- RFC 2858 – Multiprotocol Extensions for BGP-4 (June 2000)



Note: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at <http://www.ietf.org> for the latest drafts.

Before You Begin

You can configure IP multicasting on IP interfaces. For information about configuring IP interfaces, see *Chapter 2, Configuring IP*.

Enabling IP Multicasting

In this implementation, IP multicasting works on the basis of VR. By default, IP multicasting is disabled on a VR. To enable IP multicasting on a VR, use the **ip multicast-routing** command.

ip multicast-routing

- Use to enable IP multicast routing on the VR.
- By default, IP multicasting is disabled on the VR. In the disabled state, all multicast protocols are disabled, and the VR forwards no multicast packets.
- Example

```
host1(config)#ip multicast-routing
```
- Use the **no** version to disable IP multicast routing on the VR.

Deleting Multicast Forwarding Entries

You can clear one or more forwarding entries from the multicast routing table. However, if you do so, the entries may reappear in the routing table if they are rediscovered.

clear ip mroute

- Use to delete IP multicast forwarding entries.
- If you specify an asterisk (*), the system clears all IP multicast forwarding entries.
- If you specify the IP address of a multicast group, the system clears all multicast forward entries for that group.
- If you specify the IP address of a multicast group and the IP address of a multicast source, the system clears the multicast entry that matches that group and source.
- Example

```
host1:boston#clear ip mroute *
```
- There is no **no** version.

Monitoring IP Multicast Settings

To display general information about the IP multicasting configuration on the system, use the following commands:

- **show ip mroute**
- **show ip multicast protocols**
- **show ip multicast routing**

show ip mroute

- Use to display information about all or specified multicast routes.
- Specify a multicast group IP address or both a multicast group IP address and a multicast source IP address to display information about particular multicast routes.
- Use the **summary** option to see a summary rather than a detailed description.
- Use the **count** option to display the number of groups and sources.
- Use the **statistics** option to display multicast packet statistics.
- Field descriptions
 - › (S,G) – the IP addresses of the multicast source and the multicast group
 - › Uptime – length of time in *days minutes:hours:seconds* format that the (Source, Group) pair has been active
 - › Expires – length of time in *days minutes:hours:seconds* format that the (Source, Group) pair mapping will cease
 - › RPF Route – IP address and prefix of the RPF route
 - › Incoming interface – type and specifier of the incoming interface for the RPF route
 - › Neighbor – IP address of the neighbor
 - › Owner – owner of the route
 - Local – route belonging to the local interface
 - Static – static route
 - Other protocols – route established by a protocol such as RIP or OSPF
 - › Incoming Interface List – list of incoming interfaces on the router. Details include:
 - Type of interface and its specifier
 - Action that the interface takes with packets: accept or discard
 - Multicast protocol that owns the interface
 - Time that the interface has been active in this protocol, in *days minutes:hours:seconds* format
 - Time that the interface will cease to be active in this protocol, in *days minutes:hours:seconds* format
 - › Outgoing Interface List – list of outgoing interfaces on the router. Details include:
 - Type of interface and its specifier
 - Action that the interface takes with packets: forward
 - Protocol running on the interface: PIM, DVMRP, or IGMP
 - Time that the interface has been active in this protocol, in *days minutes:hours:seconds* format
 - Time that the interface will cease to be active in this protocol, in *days minutes:hours:seconds* format
 - › Counts – numbers of types of source group mappings
 - (S,G) – number of (S,G) entries
 - (*,G) – number of (*,G) entries

- Example

```
host1#show ip mroute
IP Multicast Routing Table

(S, G) uptime d h:m:s[, expires d h:m:s]
RPF route: addr/mask, incoming interface
           neighbor address, owner route-owner
Incoming interface list:
           Interface (addr/mask), State/Owner [(RPF IIF)]
Outgoing interface list:
           Interface (addr/mask), State/Owner, Uptime/Expires

(10.0.10.1, 225.1.1.1) uptime 0 00:10:31
RPF route: 10.0.10.0/24, incoming interface atm5/3.1010
           neighbor 10.0.10.8, owner Local
Incoming interface list:
           atm5/3.1010 (10.0.10.8/24), Accept/Pim (RPF IIF)
Outgoing interface list:
           atm5/1.108 (108.0.8.5/8), Forward/Pim, 0 00:02:52/never
           atm5/1.109 (107.0.8.4/8), Forward/Pim, 0 00:10:07/never
```

show ip mroute count

- Use to display information about the number of groups and sources.
- Specify a multicast group IP address or both a multicast group IP address and a multicast source IP address to display information about particular multicast route.
- Field descriptions
 - › Counts – numbers of types of source group mappings
 - (S,G) – number of (S,G) entries
 - (*,G) – number of (*,G) entries
- Example

```
host1#show ip mroute count
IP Multicast Routing Table

Counts:      2 (S, G) entries
             0 (*, G) entries
```

show ip mroute summary

- Use to display a summary of all or specified multicast routes.
- Specify a multicast group IP address or both a multicast group IP address and a multicast source IP address to display information about particular multicast routes.
- Field descriptions
 - › Group Address – IP address of the multicast group
 - › Source Address – IP address of the multicast source
 - › RPF Route – IP address and network mask of the RPF route
 - › RPF Iif – type and identifier for the incoming interface for the RPF route
 - › #Oifs – number of outgoing interfaces
 - › Counts – numbers of types of source group mappings
 - (S,G) – number of (S,G) entries
 - (*,G) – number of (*,G) entries
- Example

```
host1#show ip mroute summary
      IP Multicast Routing Table
```

Group Address	Source Address	RPF route	RPF Iif	#Oifs
224.0.1.39	52.1.1.1	51.1.1.1/32	Register IIF	0
224.0.1.40	51.1.1.1	51.1.1.1/32	loopback1	1

```
Counts:      2 (S, G) entries
            0 (*, G) entries
```

show ip multicast protocols

- Use to display information about multicast protocols enabled on the system.
- Use the **brief** keyword to display a summary of information rather than a detailed description.
- Field descriptions
 - › Protocol – name of the multicast protocol
 - › Type – mode of the multicast protocol
 - For DVMRP – dense
 - For PIM – sparse, dense, or sparse-dense
 - For IGMP – local
 - › Interfaces
 - registered – number of interfaces on which the protocol is configured
 - owned – number of interfaces that a protocol owns. If you configure only IGMP on an interface, IGMP owns the interface. However, if you configure IGMP and either PIM or DVMRP on the same interface, PIM or DVMRP owns the interface.

- › Registered interfaces – includes the following information about interfaces on which the protocol is configured
 - Types and identifiers of interfaces. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - Protocols configured on the interface and the protocol that owns the interface. If you configure only IGMP on an interface, IGMP owns the interface. However, if you configure IGMP and PIM or DVMRP on the same interface, PIM or DVMRP owns the interface.
- › Count – number of multicast protocols on the VR
- Example

```
host1:2#show ip multicast protocols
Multicast protocols:

Protocol Pim
  Type: Sparse Dense
  Interfaces: 2 registered, 2 owned
  Registered interfaces:
    atm3/1.2 (40.2.2.2/8) local Igmp owner Pim
    loopback2 (52.1.1.1/32) owner Pim
Protocol Igmp
  Type: Local
  Interfaces: 1 registered, 0 owned
  Registered interfaces:
    atm3/1.2 (40.2.2.2/8) local Igmp owner Pim
Count: 2 protocols
```

show ip multicast protocols brief

- Use to display a summary of information about multicast protocols enabled on the system.
- Field descriptions
 - › Protocol – name of the multicast protocol
 - › Registered Interfaces – number of interfaces on which the protocol is configured.
 - › Owned Interfaces – number of interfaces that a protocol owns. If you configure only IGMP on an interface, IGMP owns the interface. However, if you configure IGMP and either PIM or DVMRP on the same interface, PIM or DVMRP owns the interface.
 - › Type – mode of the multicast protocol
 - For DVMRP – dense
 - For PIM – sparse, dense, or sparse-dense
 - For IGMP – local
 - › Count – number of multicast protocols on the VR

- Example

```

host1#show ip multicast protocols brief
show ip multicast protocols brief
Protocol   Registered   Owned       Type
           Interfaces   Interfaces
-----
Pim        2            2           Sparse Dense
Icmp       1            0           Local

Count: 2 protocols

```

show ip multicast routing

- Use to display information about the status of IP multicasting on the VR
- Example

```

host1#show ip multicast routing
Multicast forwarding is enabled on this router

```

Reverse Path Forwarding

IP multicasting uses reverse path forwarding (RPF) to verify that a router receives a multicast packet on the correct incoming interface. The RPF algorithm allows a router to accept a multicast datagram only on the interface from which the router would send a unicast datagram to the source of the multicast datagram.

Figure 3-1 illustrates reverse path forwarding in a network where all routers run *dense-mode* multicasting protocols. Routers that receive a multicast datagram associated with a group for which they have no hosts return *prune* messages upstream toward the source of the datagram. Upstream routers do not forward subsequent multicast datagrams to routers from which they receive prune messages. This technique creates a *source-rooted tree* (SRT), also known as a *shortest path tree* (SPT), — a structure that connects the source of a datagram to subnetworks of a multicast group via the shortest path. For more information on dense-mode protocols, see *PIM DM*, later in this chapter.

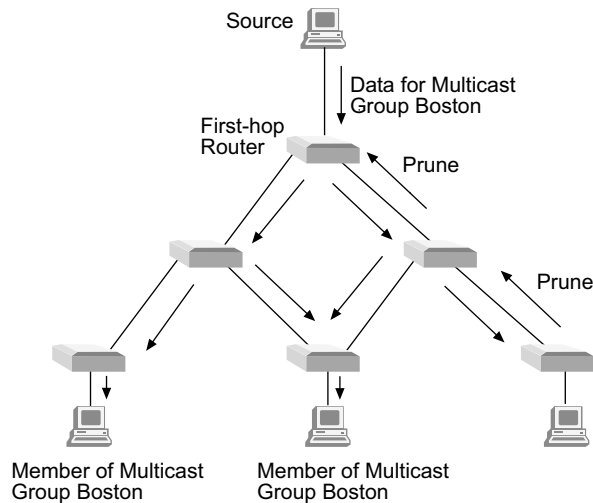


Figure 3-1 Reverse path forwarding in a dense mode environment

When all routers in a network are running *sparse-mode* multicast protocols, the routers forward a multicast datagram only to other routers with downstream members of the groups associated with the datagram. Routers running sparse-mode protocols forward multicast traffic only when explicitly requested to do so, whereas routers running dense-mode protocols forward multicast traffic except when explicitly requested not to do so. For more information on sparse-mode protocols, see *PIM SM*, later in this chapter.

RPF may take place via static routes, dynamic routes, or local subnets. You can define static routes for this purpose and view information associated with RPF routes.

Multicast Packet Forwarding

Multicast packet forwarding is based on the source (S) of the multicast packet and the destination multicast group address (G). For each $\langle S, G \rangle$ pair, the router accepts multicast packets on an incoming interface (IIF), which satisfies the RPF check (RPF-IIF). The router drops packets received on IIFs other than the RPF-IIF and notifies the routing protocols that a packet was received on the wrong interface.

The router forwards packets received on the RPF-IIF to a list of outgoing interfaces (OIFs). The list of OIFs is determined by the exchange of routing information and local group membership information. The router

maintains mappings of <S, G, IIF> to {OIF1, OIF2...} in the multicast routing table.

You can enable two or more multicast protocols on an IIF. However, only one protocol can forward packets on that IIF. The protocol that forwards packets on an IIF *owns* that IIF. A multicast protocol that owns an IIF also owns the <S,G> entry in the multicast routing table.

ip rpf-route

- Use to customize static routes that the system may use for RPF.
- Specify the IP address and subnet mask of the destination network.
- Specify either a next-hop IP address or an interface type and specifier, such as atm 3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Optionally, specify the distance (number of hops) to the next-hop address.
- Optionally, specify a route's tag number to identify a particular route in the routing table.
- Example


```
host1(config)#ip rpf-route 11.1.0.0 255.255.0.0 atm4/1.1 56
tag 25093
```
- Use the **no** version to remove the static route.

show ip rpf-route

- Use to display routes that the system can use for RPF.
- Specify the IP address and the network mask to view routes to a particular destination.
- Specify a unicast routing protocol to view routes associated with that protocol.
- Field descriptions
 - › Proto
 - Connect – subnet directly connected to the interface
 - Static – static route
 - *protocol-name* – route learned via the named protocol
 - › Prefix – value of the logical AND of the IP address of the destination network and the subnet address
 - › Length – length of the subnet mask in bits
 - › Next Hop – IP address of the next hop for this route
 - › Dist – distance configured for this route
 - › Met – learned or configured cost associated with this route
 - › Intf – type of interface and interface specifier for the next hop. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.

- Example

```
host1#show ip rpf-route
Proto:      Prefix/Length:  Next Hop:      Dist/Met:      Intf:
Connect    10.5.0.0/16    10.5.3.149     0/1            fastEthernet0/0
Static     11.0.0.0/8     21.1.1.2       1/1            atm4/0.1
Connect    21.1.1.0/24    21.1.1.2       0/1            atm4/0.1
Connect    25.25.25.25/32 25.25.25.25    0/1            loopback0
```

Using Unicast Routes for RPF

You can use the **ip route-type** command to specify that IS-IS, OSPF, or RIP routes should be available for RPF. Routes available for RPF appear in the multicast view of the routing table.

ip route-type

- Use to specify whether IS-IS, OSPF, or RIP routes are available only for unicast forwarding, only for multicast reverse path forwarding checks, or for both.
- Use the **show ip rpf-routes** command to view the routes available for RPF.
- By default, IS-IS, OSPF, and RIP routes are available both for unicast forwarding and multicast reverse path forwarding checks.

- Example

```
host1(config)#router ospf
host1(config-router)#ip route-type multicast
```

- There is no **no** version.

IGMP

IP hosts use Internet Group Management Protocol (IGMP) to report their multicast group memberships to neighboring routers. Similarly, multicast routers, such as the ERX system, use IGMP to discover which of their hosts belong to multicast groups.

The IPv4 address scheme assigns Class D addresses for IP multicasting. IGMP is the protocol that uses these addresses, which can be in the range 224.0.0.0 to 239.255.255.255. The following addresses have specific functions or are unavailable:

- 224.0.0.0 is reserved, and you cannot assign it to a group.
- 224.0.0.1 is the all-hosts address – a packet sent to this address reaches all hosts on a subnet.
- 224.0.0.2 is the all-routers address – a packet sent to this address reaches all routers on a subnet.

This implementation of IGMP complies with IGMPv2, which supports both IGMPv1 and IGMPv2 hosts.

IGMP Operation

IGMPv2 involves the exchange of the following types of messages between routers and hosts:

- Group membership queries
- Group membership reports
- Leave group membership messages

Group Membership Queries

A multicast router can be a querier or a nonquerier. There is only one querier on a network at any time. Multicast routers monitor queries from other multicast routers to determine the status of the querier. If the querier hears a query from a router with a lower IP address, it relinquishes its role to that router.

Multicast routers send two types of group membership queries to hosts on the network:

- General queries to the all-hosts group address (224.0.0.1)
- Specific queries to the appropriate multicast group address

The purpose of a membership group query is to discover the multicast groups to which a host belongs.

IGMPv2 group membership queries have a *Max Response Time* field. This response time is the maximum that a host can take to reply to a query.

Group Membership Reports

When a host receives a group membership query, it identifies the groups associated with the query and determines to which groups it belongs. The host then sets a timer, with a value less than the *Max Response Time* field in the query, for each group to which it belongs.

When the timer expires, the host multicasts a group membership report to the group address. When a multicast router receives a report, it adds the group to the membership list for the network and sets a timer to the *Group Membership Interval*. If this timer expires before the router receives another group membership report, the router determines that the group has no members left on the network.

If the router does not receive any reports for a specific multicast group within the *Max Response Time*, it assumes that the group has no members on the network. The router does not forward subsequent multicasts for that group to the network.

Leave Group Membership Messages

When a host leaves a group, it sends a leave group membership message to multicast routers on the network. A host generally addresses leave group membership messages to the all-routers group address, 224.0.0.2.

Configuring Static and Dynamic IGMP Interfaces

The system supports *static* and *dynamic* IGMP interfaces. Unlike static interfaces, dynamic interfaces are not restored when you reboot the system. For some protocols, dynamic layers can build on static layers in an interface; however, in a dynamic IGMP interface, all the layers are dynamic. See Figure 3-2 for examples of static and dynamic IGMP interfaces.

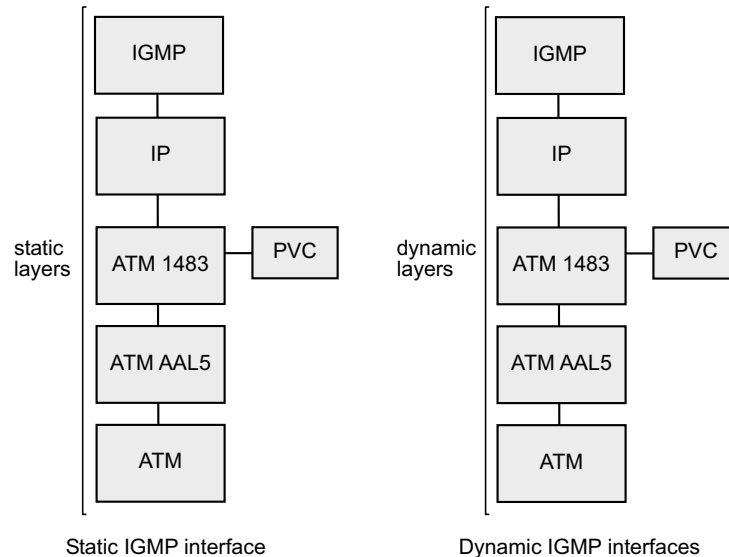


Figure 3-2 Static and dynamic IGMP interfaces

Static IGMP interfaces are configured with software such as the CLI or an SNMP application; dynamic IGMP interfaces are configured with a profile. A profile comprises a set of attributes for an interface; a profile for dynamic IGMP interfaces contains attributes for configuring all the layers in the interface.

You define a profile using the same CLI commands that you use to configure a static IGMP interface; however, the mode in which you use the commands differs. Use the commands in Interface Configuration mode to configure a static IGMP interface and in Profile Configuration mode to define a profile.

When you have defined a profile, you can apply it to an interface or group of interfaces. Profiles provide an efficient method of creating and managing large numbers of dynamic interfaces. For detailed information about creating and assigning profiles, see *ERX Physical and Link Layers Configuration Guide, Chapter 21, Configuring Dynamic Interfaces*. When you create a profile for dynamic IGMP interfaces, specify attributes for configuring all layers in the interface.

You use the following IGMP commands to configure a static IGMP interface. You also use these commands to define the attributes for the IGMP layer when you create a profile for dynamic IGMP interfaces.

- ip igmp
- ip igmp access-group
- ip igmp immediate-leave
- ip igmp last-member query-interval
- ip igmp promiscuous
- ip igmp querier
- ip igmp querier-timeout
- ip igmp query-interval
- ip igmp query-max-response-time
- ip igmp robustness
- ip igmp static-group
- ip igmp version

The following sections describe the tasks associated with these commands.

Enabling IGMP on an Interface

You must start IGMP on each interface that you want to use the protocol. You can configure IGMP and either PIM or DVMRP on the same interface. If you configure IGMP only on an interface, the system considers that IGMP “owns” that interface. If you configure IGMP and either PIM or DVMRP on an interface, the system considers that PIM or DVMRP owns the interface.

For networks that use only IGMPv1, you can configure an interface to operate in IGMPv1 mode. However, IGMPv2 interfaces will support IGMPv1 hosts. In an IGMPv1 network, you must configure one interface to act as a querier. In an IGMPv2 network, the querier is the router with the lowest IP address.

To start IGMP, complete the following steps:

- 1 Enable IGMP on the interface.

```
host1:boston(config-if)#ip igmp
```
- 2 (IGMPv1 only) Specify IGMPv1 for the interface.

```
host1:boston(config-if)#ip igmp version 1
```
- 3 (IGMPv1 only) Specify that the interface will act as the querier for the network.

```
host1:boston(config-if)#ip igmp querier
```

ip igmp

- Use to enable IGMP on an interface.
- Example

```
host1:boston(config-if)#ip igmp
```
- Use the **no** version to disable IGMP on an interface.

ip igmp querier



- Use to specify that this IGMPv1 interface will act as a querier.
Note: This command is valid only for interfaces on which you configured IGMPv1.
- Example

```
host1:boston(config-if)#ip igmp querier
```
- Use the **no** version to restore the default situation, in which the interface does not act as a querier.

ip igmp version

- Use to set the IGMP version for the interface.
- Example

```
host1:boston(config-if)#ip igmp version 1
```
- Use the **no** version to set the version to the default, IGMPv2.

Configuring IGMP Settings for an Interface

When you start IGMP on an interface, it operates with the default settings. You can, however, modify:

- The method that the router uses to remove hosts from multicast groups (IGMPv2 interfaces only).
- The time interval at which the querier multicasts group membership queries.
- The time that a querier waits before sending a new query to hosts from which it receives leave group membership messages.
- The time that a new querier waits before sending query messages after it assumes responsibility from another querier.
- The time that a host can take to reply to a query (maximum response time).
- The number of times that the system sends each IGMP messages from this interface.

ip igmp immediate-leave

- Use to specify that when the router receives a leave group membership message from a host associated with this interface, the router will immediately remove that host from the multicast group.



Caution: Issue this command only on IGMPv2 interfaces to which one IGMP host is connected. If there is more than one IGMP host connected to a LAN via the same interface, and one host sends a leave group message, the router will remove all hosts on the interface from the multicast group. The router will lose contact with the hosts that should remain in the multicast group until they send join requests in response to the router's next general group membership query.

- Example

```
host1:boston(config-if)#ip igmp immediate-leave
```
- Use the **no** version to restore the default situation, in which the router removes a host from a multicast group if that host does not return a group membership report within a certain length of time of receiving a group membership query from the router.

ip igmp last-member query-interval

- Use to specify in tenths of a second how long the system waits before sending out another query to a host that sent a leave group membership message.
- Using a lower value allows members to leave groups more quickly.
- Example

```
host1:boston(config-if)#ip igmp last-member-query-interval 90
```

- Use the **no** version to restore the default, 10-tenths of a second (1 second).

ip igmp querier-timeout

- Use to set the time in seconds that the interface waits before sending query messages after it becomes the querier.
- Example

```
host1:boston(config-if)#ip igmp querier-timeout 200
```

- Use the **no** version to set the time to the default, twice the query interval.

ip igmp query-interval

- Use to specify how often the interface sends group membership queries.
- Example

```
host1:boston(config-if)#ip igmp query-interval 100
```

- Use the **no** version to set the polling interval to the default, 125 seconds.

ip igmp query-max-response-time

- Use to specify the period in tenths of a second during which the host is expected to respond to a group membership query.
- IGMPv2 includes this value in IGMP query messages sent out on the interface.
- You cannot set this value on interfaces running IGMPv1.
- Using a lower value allows members to join and leave groups more quickly.
- Example

```
host1:boston(config-if)#ip igmp query-max-response-time 120
```

- Use the **no** version to restore the default, 10-tenths of a second (1 second).

ip igmp robustness

- Use to specify the number of times that the system sends each IGMP message from this interface.
- Use a higher value to ensure high reliability from IGMP.
- Specify a number in the range 1–4.
- Example

```
host1:boston(config-if)#ip igmp robustness 2
```

- Use the **no** version to restore the default, 3.

Assigning a Multicast Group to an Interface

You can assign an interface to send and receive all traffic for a particular multicast group. This feature allows you to control the IGMP traffic and to test the behavior of multicast protocols in the network.

ip igmp static-group

- Use to send and receive all traffic for a multicast group from a specific interface.
- The interface sets no timers for this group.
- Example

```
host1:boston(config-if)#ip igmp static-group 225.1.2.3
```
- Use the **no** version to remove the group from the interface.

Specifying Multicast Groups

You can use a standard IP access list to specify the multicast groups that a host can join.

ip igmp access-group

- Use to restrict hosts on this subnet to joining only multicast groups that appear on the specified IP access list.
- Example

```
host1:boston(config-if)#ip igmp access-group boston-list
```
- Use the **no** version to disassociate the interface from an access list and to allow hosts on the interface to join any multicast group.

Accepting IGMP Reports from Remote Subnets

By default, IGMP interfaces accept IGMP reports only from associated subnets. You can configure the system to accept IGMP reports from subnets that are not associated with its interfaces. The **igmp promiscuous** command in Router Configuration mode specifies whether or not interfaces on the router should accept IGMP reports from indirectly connected subnets. To override this global setting on a particular interface, use the **ip igmp promiscuous** command in Interface Configuration mode.

Example In the following example, the router is configured to accept IGMP reports from indirectly connected subnets on all interfaces. The interface on port 0 of the line module in slot 4 is then configured to accept IGMP reports only from directly connected subnets.

```
host1(config)#virtual-router boston
host1:boston(config)#router igmp
host1:boston(config-router)#igmp promiscuous
```

```
host1:boston(config-router)#exit
host1:boston(config)#interface serial 4/0
host1:boston(config-if)#ip igmp promiscuous off
```

igmp promiscuous

- Use to allow all IGMP interfaces on the router to accept IGMP reports from hosts on any subnet.

- Example

```
host1:boston(config-router)#igmp promiscuous
```

- Use the **no** version to allow IGMP interfaces on the router to accept IGMP reports only from hosts on their associated subnets.

ip igmp promiscuous

- Use to specify whether the interface should accept IGMP reports from hosts on any subnet.

- › Use the **on** keyword to enable the interface to accept IGMP reports from hosts on any subnet.

- › Use the **off** keyword to allow the interface to accept IGMP reports only from hosts on subnets associated with this interface

- Example

```
host1:boston(config-if)#ip igmp promiscuous on
```

- Use the **no** version to configure an IGMP interface to use the Router Configuration mode setting to determine the subnets from which it can accept IGMP reports.

Disabling and Removing IGMP

You can disable and reenable IGMP on the VR. You can also remove IGMP from the VR and recreate it on the VR.

igmp disable

- Use to disable IGMP on a VR.

- Example

```
host1(config)#virtual-router boston
host1:boston(config)#router igmp
host1:boston(config-router)#igmp disable
```

- Use the **no** version to enable IGMP on a VR.

router igmp

- Use to create and enable IGMP on a VR or to access IGMP Router Configuration mode.
- Example

```
host1(config)#virtual-router boston
host1:boston(config)#router igmp
```
- Use the **no** version to delete IGMP and IGMP proxy from the VR.

Monitoring IGMP

You can establish a reference point for IGMP statistics by setting the statistics' counters to zero.

To display IGMP parameters, use the **show** commands described in this section.

baseline ip igmp

- Use to set the counters for IGMP statistics to zero.
- Example

```
(host1)#baseline ip igmp
```
- There is no **no** version.

show ip igmp

- Use to display IGMP information for a VR.
- Field descriptions
 - › Administrative State – status of IGMP in the software: enabled or disabled
 - › Operational State – status of IGMP on the VR: enabled or disabled
 - › Total Interfaces – number of interfaces on which you started IGMP
 - › Enabled – number of interfaces on which IGMP is enabled
 - › Disabled – number of interfaces on which IGMP is disabled
 - › Learnt Groups – number of multicast groups that the VR has discovered
 - › IGMP Statistics: Rcvd – statistics for IGMP messages received
 - Total – number of IGMP messages received
 - Checksum Errors – number of IGMP messages received with checksum errors
 - Unknown Types – number of messages received that are not group membership queries, group membership reports, or leave group membership messages
 - Queries – number of group member queries
 - Reports – number of group membership reports
 - Leaves – number of leave group membership messages
 - › IGMP Statistics: Sent – number of group member queries sent

```

host1:boston#show ip igmp
Routing Process IGMP, Administrative state enabled,
Operational state enabled
    2 total interfaces, 2 enabled, 0 disabled
    2 learnt groups
IGMP Statistics:
    Rvcd: 1 total, 0 checksum errors, 0 unknown types
        0 queries, 1 reports, 0 leaves
    Sent: 11 total

```

show ip igmp groups

- Use to display statically joined and directly connected groups learned via IGMP.
- Field descriptions
 - › Grp Address – address of the multicast group
 - › Interface – interface that discovered the multicast group
 - › State – IGMP version on the interface
 - › ExpTim – time, in seconds, at which the router decides there are no more members of this group
 - › v1HTim – time at which the router decides there are no more IGMPv1 members of a group. If this value is 0, the interface has received no IGMPv1 reports for the group.
- Example

```
host1:boston#show ip igmp groups
```

Grp Address	Interface	State	ExpTim	v1HTim
225.1.1.1	fastEthernet0/0	Version2	never	0
232.1.1.1	fastEthernet0/0	Version2	359	0

```
Count: 2 Groups
```

```
(Note: 225.1.1.1 is a "static group")
```

show ip igmp interface

- Use to display IGMP information for interfaces on which you enabled IGMP.
- Specify the **brief** keyword to see a summary of the information.
- Specify the **count** keyword to see the number of IGMP interfaces.
- Specify the **group** address keyword to see information for interfaces that belong to that group.
- Field descriptions
 - › Interface – type of interface and interface specifier. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Address – IP address of the interface
 - › Administrative state – status of the interface in the software: enabled or disabled

- › Operational state – physical status of the interface: enabled or disabled
 - › Version – IGMP version
 - › State – function of the interface: querier or nonquerier
 - › Query Interval – time interval at which this interface sends query messages
 - › Other querier present interval – time that the interface waits before declaring itself as the querier
 - › Maximum response time – time interval during which this interface expects a host to respond
 - › Last member query interval – time that this interface waits before sending a new query to a host that sends a group leave message
 - › Robustness – number of times this interface sends IGMP messages
 - › Interface defaults to global promiscuous mode – interface uses the setting of the **igmp promiscuous** command to determine whether it accepts IGMP reports from hosts on any subnet
 - › No inbound access group – no access list specified with the **ip igmp access-group command**
 - › Immediate Leave – setting of the **ip igmp immediate-leave command**: enabled or disabled
 - › Interface statistics: Rcvd – information about IGMP messages received on this interface
 - Reports – number of group membership reports received
 - Leaves – number of group leave messages received
 - Wrong Version Queries – number of group membership queries received from devices running a different version of IGMP
 - › Interface statistics: Sent – number of IGMP messages this interface has sent
 - › Interface statistics: Groups learned – number of groups this interface has discovered
 - › Count – total number of IGMP interfaces
- Example

```
host1:boston#show ip igmp interface
Interface ATM2/1.15 address 15.0.0.2/255.255.255.0
Administrative state enabled, Operational state enabled
Interface parameters:
  Version 2
  State Querier
  Query Interval 125 secs, 53 secs before the next query
  Other querier present interval 250 secs
  Maximum response time 100 (in 10ths of a second)
  Last member query interval 10 (in 10ths of a second)
  Robustness 3
  Interface defaults to global promiscuous mode
  No inbound access group
  Immediate Leave: disabled
```

```

Interface statistics:
  Rcvd: 0 reports, 0 leaves, 0 wrong version queries
  Sent: 1 queries
  Groups learned: 1

Counts: 0 down, 0 init state, 1 querier, 0 non-querier,
        1 Total

```

show ip igmp interface brief

- Use to display a summary of IGMP information for interfaces on which you enabled IGMP.
- Field descriptions
 - › Interface – type of interface and interface specifier. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Intf Address – IP address of the interface
 - › Ver – IGMP version
 - › State – function of the interface: querier or nonquerier
 - › Querier – IP address of the querier on the network to which this interface connects
 - › QTime – time interval at which this interface sends query messages
 - › QPTime – time that the interface waits before declaring itself as the querier
- Example

```

host1:boston#show ip igmp interface brief
Interface          Intf Address      Ver  State    Querier          QTime  QPTime
-----
fastEthernet0/0   192.168.1.250/24  2    Querier  192.168.1.250   28     0
atm3/0.2          21.1.1.1/8       2    Querier  21.1.1.1        26     0
Count: 2 interfaces

```

IGMP Proxy

IGMP proxy enables the system to issue IGMP host messages on behalf of hosts that the system discovered via standard IGMP interfaces. The system acts as a *proxy* for its hosts.

Overview

Figure 3-3 shows a system in an IGMP proxy configuration. You enable IGMP proxy on one interface, which connects to a router closer to the root of the tree. This interface is the *upstream interface*. The router on the upstream interface should be running IGMP.

You enable IGMP on the interfaces that connect the system to its hosts that are farther away from the root of the tree. These interfaces are known as *downstream interfaces*.

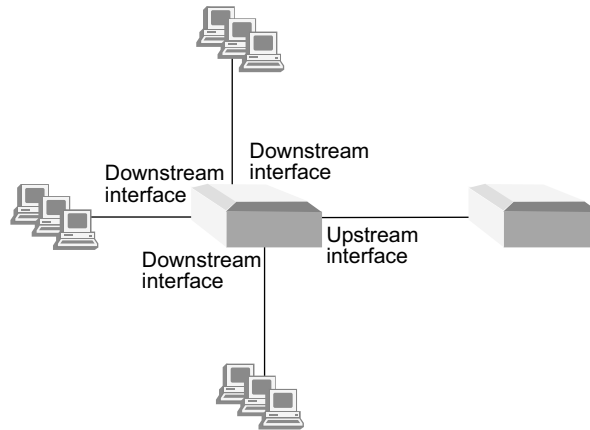


Figure 3-3 Upstream and downstream interfaces

As described in *IGMP Operation*, earlier in this chapter, hosts interact with the system through the exchange of IGMP messages. Similarly, when you configure IGMP proxy, the system interacts with the router on its upstream interface through the exchange of IGMP messages. However, when acting as the proxy, the system performs the host portion of the IGMP task on the upstream interface as follows:

- When queried, sends group membership reports to the group.
- When one of its hosts joins a multicast address group to which none of its other hosts belong, sends unsolicited group membership reports to that group.
- When the last of its hosts in a particular multicast group leaves the group, sends an unsolicited leave group membership report to the all-routers group (244.0.0.2).

Configuring IGMP Proxy

To configure a downstream interface, enable IGMP on that interface. To configure IGMP proxy on the system, complete the following tasks:

- 1 Enable IP multicasting.

```
host1(config)#ip multicast-routing
```
- 2 Identify the interface that you want to act as the upstream interface.

```
host1(config-if)#interface atm 3/0
```

- 3 Enable IGMP proxy on that interface.

```
host1(config-if)#ip igmp-proxy
```

- 4 (Optional) Specify how often the system should send unsolicited reports to routers on the upstream interface.

```
host1(config-if)#ip igmp-proxy unsolicited-report-interval
600
```

- 5 (Optional) Specify how long the system should assume that there is an IGMPv1 querier router on the subnet after the system receives an IGMP V1 query on this interface.

```
host1(config-if)#ip igmp-proxy V1-router-present-time 600
```

ip igmp-proxy



- Use to enable IGMP proxy on an interface.
- **Note:** *You can enable only one upstream interface.*
- The interface for which you enable IGMP proxy is the upstream interface.
- Example

```
host1(config-if)#ip igmp-proxy
```

- Use the **no** version to disable IGMP proxy on an interface.

ip igmp-proxy unsolicited-report-interval



- Use to specify how often the upstream interface should transmit unsolicited reports.
- **Note:** *Issue this command only on the upstream interface. Otherwise, this command will have no effect.*
- Example

```
host1(config-if)#ip igmp-proxy unsolicited-report-interval
600
```

- Use the **no** version to transmit unsolicited reports using the default value, 400 seconds.

ip igmp-proxy V1-router-present-time



- Use to specify how long the system assumes that there is an IGMPv1 querier router on the subnet after the system receives an IGMP V1 query on this interface.
- **Note:** *Issue this command only on the upstream interface. Otherwise, this command will have no effect.*
- Example

```
host1(config-if)#ip igmp-proxy V1-router-present-time 600
```

- Use the **no** version to set the time to the default, 10 seconds.

Setting the IGMP Proxy Baseline

You can set the counters for the numbers of queries received and reports sent on the upstream interface to zero. This feature allows you to establish a reference point for IGMP proxy statistics.

baseline ip igmp-proxy interface



- Use to set the counters for the numbers of queries received and reports sent on the upstream interface to zero.

Note: Issue this command only on the upstream interface. Otherwise, this command will have no effect.

- Example

```
(host1)#baseline ip igmp-proxy interface
```

- There is no **no** version.

Monitoring IGMP Proxy

To display IGMP proxy parameters, use the following **show** commands.

show ip igmp-proxy

- Use to display IGMP proxy parameters for a VR.
- Field descriptions
 - › Routing Process – IGMP proxy protocol
 - › Administrative state – state of IGMP proxy in the software
 - › Operational state – operational state of IGMP proxy: enabled or disabled
 - › Total interfaces – number of IGMP proxy interfaces on the VR; currently only one upstream interface per VR
 - › State – operational state of the IGMP proxy interfaces: up or down
 - › Multicast group – number of multicast groups associated with IGMP proxy interfaces
- Example

```
host1#show ip igmp-proxy
Routing Process IGMP Proxy, Administrative state enabled,
Operational state enabled
total 1 upstream interface, state enabled
6 multicast group
```

show ip igmp-proxy groups

- Use to display information about multicast groups that IGMP proxy reported.
- Field descriptions
 - › Grp Address – address of the multicast group
 - › Interface – type and identifier of the upstream interface associated with the multicast group
 - › Member State
 - Idle – interface is going to send a group membership report to respond to a group membership query for this group
 - Delay – interface has responded to the latest group membership query for this group
 - › Count – total number of multicast groups associated with this interface
- Example 1

```
host1#show ip igmp-proxy groups
```

Grp Address	Interface	Member State
225.1.1.1	atm3/0.2	Idle
225.1.1.2	atm3/0.2	Idle
225.1.1.3	atm3/0.2	Idle
225.1.1.4	atm3/0.2	Idle
225.1.1.5	atm3/0.2	Idle
225.1.1.6	atm3/0.2	Idle
count 6		

- Example 2

```
host1#show ip igmp-proxy group 225.1.1.1
```

Grp Address	Interface	Member State
225.1.1.1	atm3/0.2	Idle

- Example 3

```
host1#show ip igmp-proxy group count
```

```
Count: 6 groups
```

show ip igmp-proxy interface

- Use to display information about the interface on which you configured IGMP proxy.
- To view information about a particular interface, enter an interface type and specifier, such as atm 3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Specify the **brief** option to display a summary rather than a detailed description.

- Field descriptions
 - › Interface – type of upstream interface. For details about interface types, see *ERX Command Reference Guide, About This Guide*.
 - › Address – address of upstream interface
 - › Administrative state – state of upstream interface in the software: enabled or disabled
 - › Operation state – physical state of upstream interface: enabled or disabled
 - › Version – IGMP version on this interface
 - › State – presence of IGMPv1 routers on the same subnet as this upstream interface
 - › Unsolicited report interval – time interval at which this upstream interface sends unsolicited group membership report
 - › Version 1 router present timeout – how long the upstream interface assumes there is an IGMPv1 router on the subnet after that interface receives an IGMPv1 group membership query
 - › multicast group – number of multicast groups associated with this upstream interface
 - › Interface statistics: Rcvd – statistics for messages received on this interface
 - v1 queries – number of IGMPv1 group membership queries received
 - v2 queries – number of IGMPv2 group membership queries received
 - v1 report – number of IGMPv1 group membership reports received
 - v2 report – number of IGMPv2 group membership reports received
 - › Interface statistics: Sent – statistics for messages sent from this interface
 - v1 reports – number of IGMPv1 leave group reports sent
 - v2 reports – number of IGMPv2 leave group reports sent
 - leaves – number of leave group membership messages sent
- Example

```
host1#show ip igmp-proxy interface atm 3/0.2
Interface atm3/0.2 address 21.1.1.1/255.0.0.0
Administrative state enabled, Operational state enabled
Interface parameters:
  Version 2
  State No v1 Router Present
  Unsolicited report interval 10 secs
  Version 1 router present timeout 400 secs
  0 multicast group
Interface statistics:
  Rcvd:  0 v1 query, 6 v2 queries
         0 v1 report, 0 v2 report
  Sent:  0 v1 report, 48 v2 reports, 0 leave
```

PIM

Protocol Independent Multicast (PIM) is the protocol that allows multicast routers to identify other multicast routers that should receive packets. This implementation of PIM supports PIM Dense Mode (PIM DM), PIM Sparse Mode (PIM SM), and PIM Sparse-Dense Mode (PIM S-DM).

Figure 3-4 represents how PIM builds a source-group entry in an SRT. When multiple routers are connected to a multiaccess network, one router must assume the role of the designated router (DR). The DR receives data from the source on interface 1/0 and multicasts the data to its downstream neighbors on interfaces 1/1, 2/0, and 2/1. In the DR routing table, the entry for this operation lists the source as the IP address of the source and the group as the IP address of the multicast group.

Neighbors exchange hello messages periodically to determine the DR. The router with the highest network layer address assumes the role of the DR. If the DR subsequently receives a hello message from a neighbor with a higher network layer address, that neighbor becomes the DR.

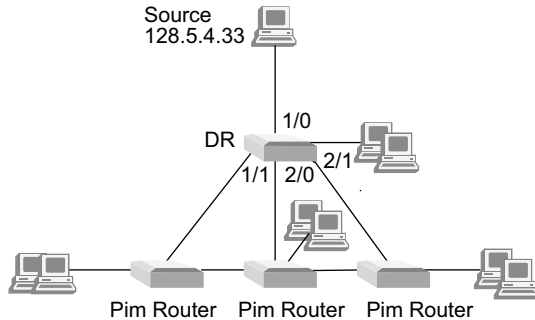


Figure 3-4 Source-rooted tree

DR Routing Table Entry

Source	128.5.4.33
Group	225.1.3.5
Register	1/0
RP	1/1, 2/0, 2/1
Input Interface	1/0
Output Interface	1/1, 2/0, 2/1

PIM DM

PIM DM uses a reverse path multicast, flood and prune mechanism. The protocol was developed for situations that meet one or more of the following criteria:

- Sources and receivers are close together, and there are many more receivers than senders.
- There is a constant stream of multicast data.
- There is a lot of multicast data.

Dense-mode routing protocols use SRT algorithms. An SRT algorithm establishes a tree that connects each source in a multicast group to the members of the group. All traffic for the multicast group passes along this tree.

Figure 3-5 illustrates how PIM DM works. When a source sends a multicast packet to a first-hop router, the first-hop router multicasts that packet to its neighbors. Those neighbors in turn forward the packet to their neighbors and their hosts that belong to the multicast group. If a neighbor has no hosts that belong to the multicast group and has no other PIM neighbors, it returns a prune message to the first-hop router. The first-hop router does not multicast subsequent packets for that group to neighbors who respond with prune messages.

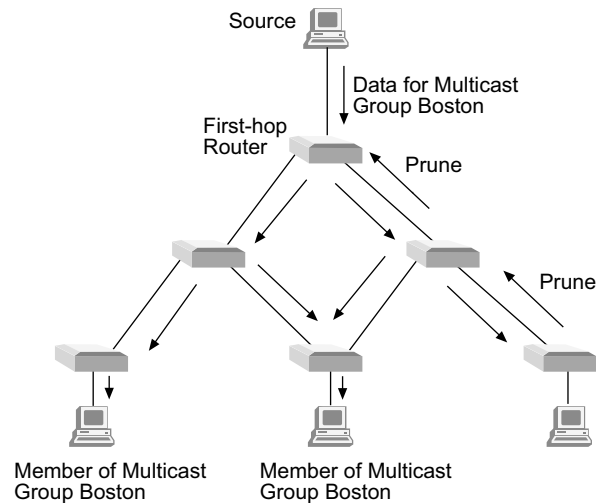


Figure 3-5 PIM DM operation

Overriding Prunes

If a host on a previously pruned branch wants to join a multicast group, it sends an IGMP message to its first-hop router. The first-hop router then sends a graft message upstream.

PIM routers send join messages on multiaccess interfaces to override prune messages. For example, if a PIM router sent a prune message to indicate that it had no hosts for a multicast group, and one of its hosts subsequently wants to send a packet to that group, the router sends a join message to the first-hop router.

Preventing Duplication

If there are parallel paths to a source, duplicate packets can travel via different routers downstream to the network. If a forwarding router receives a multicast packet on its outgoing interface, the router knows that the packet is a duplicate and notifies the upstream routers. See Figure 3-6.

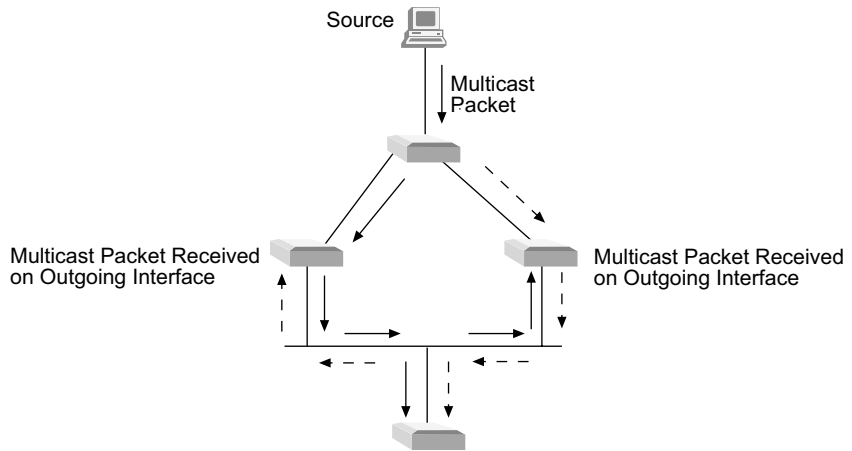


Figure 3-6 Detecting duplication

The upstream routers responsible for the duplication send assert messages to determine which router should be the forwarder. Downstream routers listen to the assert messages to discover which router becomes the forwarder.

PIM SM

This implementations of PIM-SM supports the following features:

- Rendezvous point (RP) routers
- DRs and DR election
- Join/prune messages, hello messages, assert messages, register messages
- Switching from a shared tree to an SPT
- (*,*,RP) support for interoperation with dense-mode protocols
- RPF checks of multicast entries when unicast routing configuration changes
- Timers for tree maintenance
- Border, null, RPT, SPT, and wildcard flags

- Remote neighbors

PIM SM was developed for situations that meet one or more of the following criteria:

- The multicast group contains few receivers.
- Multicast traffic is infrequent.
- WANs separate sources and receivers.

Sparse-mode routing protocols use *shared trees*. In a shared tree, sources forward multicast datagrams to a directly connected router, the DR. The DR encapsulates the datagram and unicasts it to an assigned RP router, which then forwards the datagram to members of multicast groups.

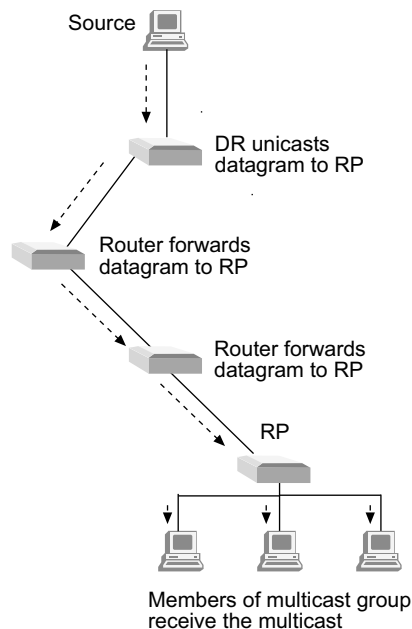


Figure 3-7 PIM SM operation

In PIM SM, an RP announces a source and establishes paths from the source to members of a multicast group before multicasting any datagrams. RPs transmit join messages to become part of the shared tree that allows distribution of packets to the multicast group.

However, when a source starts multicasting datagrams, PIM SM can switch to an SRT—known in PIM SM as an SPT—to improve the network's efficiency. Although shared trees minimize the traffic in the network and the costs associated with unnecessary transmission of data,

the routes in a shared tree may be longer than those in an SPT. See Figure 3-8.

The DRs on the network determine when the source switches from a shared tree to an SPT. A DR switches to the SPT when it receives a certain number of packets, which you can configure using the **ip pim-spt threshold** command. This command has a default value of zero, which causes a DR to switch to an SPT immediately after it receives its first multicast data packet.

When all DR routers associated with a specific RP router have switched to the SPT, the RP router sends a join/prune message toward the multicast source. When the multicast source receives this message, it starts sending multicast data via the SPT.

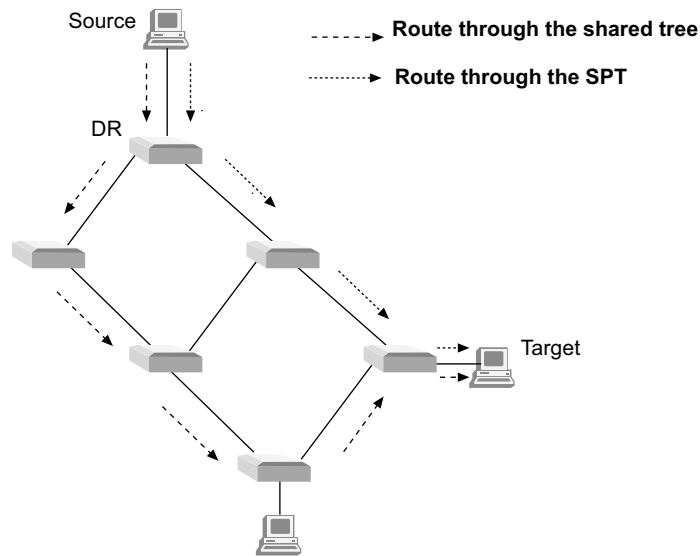


Figure 3-8 Shared tree versus SPT

Joining Groups

A host's DR sends join messages to the RP when that host wants to join a group. When a host wants to leave a group, it communicates with its DR via IGMP. When the DR no longer has any hosts that belong to a particular group, it sends a prune message to the RP.

Remote Neighbors

You can create remote neighbors in PIM SM. This feature enables an ERX system to establish neighbor adjacencies with other ERX systems through a pair of unidirectional interfaces, such as the endpoints of an MPLS tunnel. Figure 3-9 shows an example in which two ERX systems, called boston and chicago, are remote neighbors connected by two unidirectional MPLS tunnels.

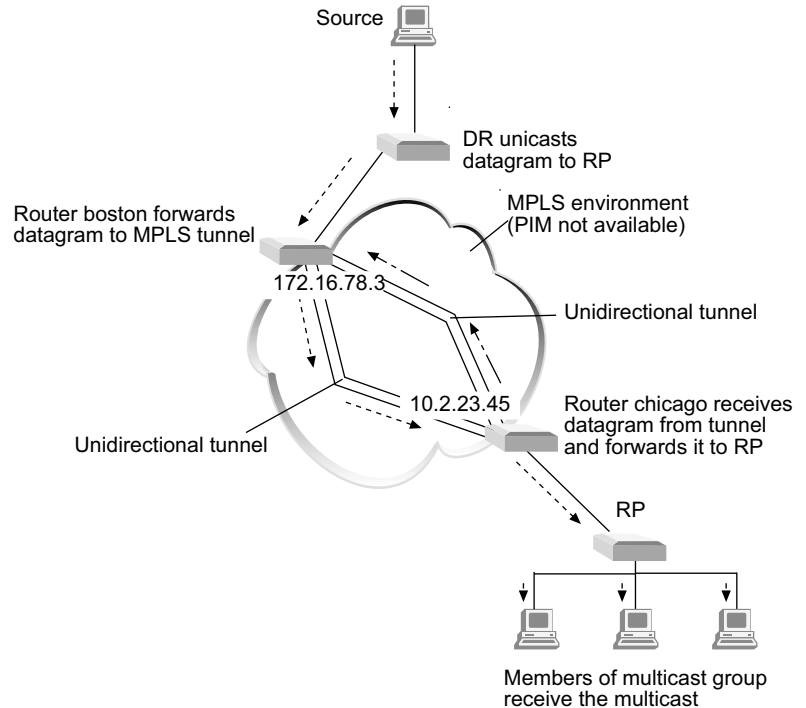


Figure 3-9 PIM remote neighbors connected by MPLS tunnel

On each ERX system, you must specify the location of the interface that PIM uses as the source address for the connection *to* the remote neighbor. You must also specify that the other system is a remote neighbor, and identify the IP address of the other system that PIM uses as the source address for the connection *from* the remote neighbor.

Bear the following issues in mind when configuring remote neighbors:

- A route to the source RP must exist in the unicast view of the routing table to ensure that the PIM router can detect the remote neighbor. For information about configuring routes in the unicast routing tables, see the chapters on unicast routing protocols in this book and *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 1, Configuring BGP Routing*.
- A route to the source must be present in the multicast view of the IP routing table to ensure that the PIM router can perform an RPF check for that source in a VPN.

To add a route to the multicast IP routing table across a VPN, you can:

- > Configure a multicast static route by issuing the **ip rpf-route** command (see *Multicast Packet Forwarding*, earlier in this chapter).
- > Configure BGP to add its unicast routes to the multicast IP routing table by issuing the **ip route-type both** command. (see *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 1, Configuring BGP Routing*)
- > Configure OSPF or RIP to learn the route via their remote neighbor features (see *Chapter 6, Configuring RIP* and *Chapter 7, Configuring OSPF*).

If a route is more specific than the route used to reach the remote neighbor originally, OSPF and RIP do not insert that route in the unicast and multicast views of the IP routing table. This feature prevents OSPF and RIP from masking the original route. If you require PIM to use such a route to reach a remote neighbor, add that route to the multicast view of the IP routing table using one of the methods described in the preceding paragraphs.

Timers

PIM SM uses timers to maintain the networking trees. PIM SM routers poll their neighbors and hosts for various pieces of information at set intervals. If a PIM SM router does not receive information from a neighbor or host within a specific time, known as the *holdtime*, it removes the associated information from its routing tables.

You can configure how often an interface sends hello messages (hello-interval) and how often routers send RP announce messages (RP-Announce-Interval). The holdtime associated with hello messages is 3.5 times the hello-interval and the holdtime associated with RP announce messages is 2.5 times the RP-Announce-Interval.

All other timers are fixed and take the default values recommended in:

Protocol Independent Multicast MIB for IPv4 –
draft-ietf-idmr-pim-mib-10.txt (July 2000 expiration)

PIM S-DM

In PIM S-DM, if an RP is not known for a group, the system sends data using PIM DM. However, if the system discovers an RP or you configure an RP statically, PIM SM takes over.

You can configure both PIM DM and PIM SM commands in PIM S-DM.

Enabling and Disabling PIM on a VR

By default, PIM is disabled. To enable PIM on a VR:

- 1 Enable multicast routing.
`host1(config)#ip multicast-routing`
- 2 Create a VR, or access the VR context.
`host1(config)#virtual-router boston`
- 3 Create and enable PIM processing.
`host1:boston(config)#router pim`

To disable PIM processing on a router, use the **pim disable** command.

pim disable

- Use to disable PIM processing. By default, PIM processing is enabled.
- Example
`host1:boston(config-router)#pim disable`
- Use the **no** version to reenabling PIM processing.

router pim

- Use to create and enable PIM processing on a VR or to access Router Configuration mode for PIM.
- Example

```
host1:boston(config)#router pim
```
- Use the **no** version to remove PIM from the VR.

Enabling PIM on an Interface

You can enable PIM on an interface in one of the allowed modes and specify how often the interface should send hello messages to neighbors.

You can configure PIM and IGMP on the same interface. If you configure IGMP and PIM on an interface, the system considers that PIM owns the interface.



Note: You cannot configure DVMRP and PIM on the same interface.

ip pim

- Use to enable PIM on an interface.
- Example

```
host1(config-if)#ip pim sparse-dense-mode
```
- Use the **no** version to disable PIM on an interface.

ip pim query-interval

- Use to specify how often the router should send hello messages to neighbors.
- Example

```
host1(config-if)#ip pim query-interval
```
- Use the **no** version to restore the default setting, 30 seconds.

Configuring an RP Router for PIM SM and PIM S-DM

When you use the system for PIM SM or PIM S-DM, some VRs must act as RP routers. You can configure static RP routers or configure the system to assign RP routers automatically.

To configure the system to assign RP routers automatically, you must define several VRs as RP routers and one VR as an RP mapping agent. RP routers send their announcement messages to the RP mapping agent, which assigns groups to RP routers and resolves any conflicts. The RP mapping agent notifies neighbors of the RP assigned to each group.

Configuring a Static RP Router

If you want to control PIM more tightly, you can configure a static RP router. To do so:

- 1 Configure an access list that details the multicast groups that will use the static RP router.

```
host1(config)#access-list boston permit 224.0.0.0  
15.255.255.255
```

- 2 Specify a static RP router.

```
host1(config)#ip pim rp-address 122.0.0.1 1
```

Configuring an Auto-RP Router for PIM SM

Two multicast groups, 224.0.1.39 and 224.0.1.40, are reserved for forwarding auto-RP messages through the network. When you configure an auto-RP router for PIM SM, you must assign a static RP router to these two groups. You can then specify an RP mapping agent for other multicast groups.

To configure an auto-RP router for PIM SM:

- 1 Configure a static RP to have priority over the auto-RP router for the groups that send auto-RP multicast messages.

```
host1(config)#access-list 11 permit 224.0.1.39 0.0.0.0  
host1(config)#access-list 11 permit 224.0.1.40 0.0.0.0  
host1(config)#ip pim rp-address 192.48.1.22 76 override
```

- 2 Assign an RP mapping agent.

```
host1(config)#ip pim send-rp-discovery scope 23 loopback 1
```

- 3 Configure routers to send auto-RP announcement messages to the mapping agent.

```
host1(config)#ip pim send-rp-announce loopback 2 scope 16  
group-list 1
```

Configuring an Auto-RP Router for PIM S-DM

In PIM S-DM mode, you must prevent routers from advertising auto-RP messages to the multicast groups 224.0.1.39 and 224.0.1.40, which are reserved for forwarding auto-RP messages through the network. To configure an auto-RP router for PIM S-DM:

- 1 Assign an RP mapping agent.

```
host1(config)#ip pim send-rp-discovery scope 23 loopback 1
```

- 2 Configure an access list that details the multicast groups that will use the static RP router.

```
host1(config)#access-list boston permit 224.0.0.0
15.255.255.255
```

- 3 Prevent routers from advertising auto-RP messages to the multicast groups that are reserved for forwarding auto-RP messages through the network.

```
host1(config)#access-list 1 deny 224.0.1.39
host1(config)#access-list 1 deny 224.0.1.40
```

- 4 Configure routers to send auto-RP announcement messages to the mapping agent.

```
host1(config)#ip pim send-rp-announce loopback 2 scope 23
group-list boston interval 200
```

ip pim rp-address

- Use to specify a static PIM RP router.
- Specify a standard IP access list of multicast groups to control which multicast groups should use this RP router.
- Specify the **override** keyword if you want this static RP router to have priority over auto-RP routers.
- Example


```
host1(config)#ip pim rp-address 192.48.1.22 76 override
```
- Use the **no** version to clear the filter from this interface.

ip pim send-rp-announce

- Use to send auto-RP announcement messages from a VR configured as an RP.
- Specify an interface type and specifier, such as atm 3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*. The auto-RP announcement messages will contain the IP address for the interface you specify.
- Specify the number of hops for which the message is valid. The default is 64.
- Specify an access list that details those multicast group to stipulate which multicast groups the RP should include in announcement messages.
- Specify a time interval to control how often the system sends announcements.
- Example


```
host1(config)#ip pim send-rp-announce loopback 2 scope 23
group-list boston interval 200
```
- Use the **no** version to clear the filter from this interface.

ip pim send-rp-discovery scope

- Use to configure the system as an RP mapping agent, which records RP-to-group mappings and notifies PIM DRs about the mappings.
- Specify the number of hops for which the RP discovery message is valid. The default is 64.
- To assign an interface from which the system should send auto-RP discovery messages, specify an interface type and specifier, such as atm 3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Example

```
host1(config)#ip pim send-rp-discovery scope 23 loopback 1
```
- Use the **no** version to stop the system from acting as an RP mapping agent.

Switching to an SPT for PIM SM

PIM SM initiates multicasting using a shared tree. You can configure PIM SM to switch to an SPT when a source starts sending multicast messages or you can prevent PIM SM from switching to an SPT. Multicasting over an SPT may be more efficient than multicasting over a shared tree (see *PIM SM*, earlier in this chapter).

ip pim spt-threshold

- Use to specify when PIM SM switches from a shared tree to an SPT.
- Specify a nonzero integer or the keyword **infinity** to prevent PIM SM from switching to an SPT.
- Specify a value of 0 to configure PIM to switch to an SPT when a source starts sending multicast messages.
- Example

```
host1(config)#ip pim spt-threshold 4
```
- Use the **no** version to restore the default, 0.

Configuring PIM SM Remote Neighbors

To configure a pair of ERX systems to act as PIM remote neighbors:

- 1 On one system, specify that the other system will be a remote neighbor, and identify the IP address of the interface on the other system that is used for the connection to this system.

```
host1(config-router):boston#remote-neighbor 10.2.23.45  
sparse-mode
```

- 2 Specify the location of the local interface whose address is used as the source address for the PIM connection to a remote neighbor.

```
host1(config-router-rn):boston#update-source atm 2/1.108
```

- 3 (Optional) Specify how often the system should send hello messages to the remote neighbor.

```
host1(config-router-rn):boston#query-interval 40
```

- 4 Repeat steps 2 to 3 for the other system.

query-interval

- Use to specify how often the router should send hello messages to remote neighbors.
- Example

```
host1(config-router-rn)#ip pim query-interval 40
```

- Use the **no** version to restore the default, 30 seconds.

remote-neighbor

- Use to specify a remote neighbor for PIM sparse mode.
- Specify the IP address of the interface on the remote neighbor that PIM uses as the source address for the connection to this system.
- Example

```
host1(config-router)#remote-neighbor 10.25.100.14  
sparse-mode
```

- Use the **no** version to remove the remote neighbor and any attributes configured for the remote neighbor.

update-source

- Use to specify the PIM interface whose local address is used as the source address for the PIM connection to a remote neighbor.
- You can use the same source address to form neighbor adjacencies with more than one PIM remote neighbor.
- You must use the IP address of this interface when issuing the **remote-neighbor** command on the remote neighbor.
- Example

```
host1(config-router-rn)#update-source loopback 5
```

- Use the **no** version to delete the source address from the connection to the remote neighbor.

Configuration Example

This example uses the configuration shown in Figure 3-9. Two ERX systems called router boston and router chicago are running PIM and are

connected by MPLS tunnels. To configure the systems as PIM remote neighbors:

- 1 Specify that router *chicago* will be a remote neighbor of router *boston*, and identify the IP address on router *chicago* which will transmit datagrams to router *boston*.

```
boston(config-router)#remote-neighbor 10.2.23.45 sparse-mode
```

- 2 Specify the location of the interface that will transmit datagrams from router *boston* to router *chicago*.

```
boston(config-router-rn)#update-source atm 2/1.108
```

- 3 Specify that router *boston* will send hello messages to router *chicago* every 40 seconds.

```
boston(config-if)#ip pim query-interval 40
```

- 4 Specify that router *boston* will be a remote neighbor of router *chicago*, and identify the IP address on router *boston* that will transmit datagrams to system *chicago*.

```
chicago(config-router)#remote-neighbor 172.16.78.3  
sparse-mode
```

- 5 Specify the location of the interface that will transmit datagrams from router *chicago* to router *boston*.

```
chicago(config-router-rn)#update-source atm 2/1.95
```

- 6 Specify that router *chicago* will send hello messages to router *boston* every 40 seconds.

```
chicago(config-if)#ip pim query-interval 40
```

Removing PIM

To remove PIM from a VR, use the **no router pim** command.

router pim

- Use to create and enable PIM processing on a VR or to access Router Configuration mode.

- Example

```
host1:boston(config)#router pim
```

- Use the **no** version to remove PIM from the VR.

Resetting PIM Counters and Mappings

You can use the **clear ip pim** commands to reset PIM counters and mappings.

clear ip pim auto-rp

- Use to clear the group-to-RP router mappings the system learned through auto-RP.
- Specify the IP address of an RP to clear the group-to-RP mappings for a particular RP. If you do not specify an IP address, the system clears the group-to-RP mappings on all RP routers learned through auto-RP.

- Example

```
host1(config)#clear ip pim auto-rp 192.34.56.7
```

- There is no **no** version.

clear ip pim interface count

- Use to clear the counters for multicast packet statistics on all interfaces or a specified interface.
- Specify an interface type and identifier, such as atm 3/0 to clear the counters on that interface. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- If you do not specify an interface, the system clears the counters on all interfaces.

- Example

```
host1(config)#clear ip pim interface atm 3/0.5 count
```

- There is no **no** version.

clear ip pim remote-neighbor count

- Use to clear the counters for remote neighbor statistics on all interfaces or the specified interface.
- Specify the IP address of an interface to clear the counters for that interface.
- If you do not specify an interface, the system clears the counters on all interfaces.

- Example

```
host1(config)#clear ip pim remote-neighbor 10.2.5.8 count
```

- There is no **no** version.

Monitoring PIM

You can display information about PIM events and parameters.

Monitoring PIM Events

You can use the debug PIM commands to view information about PIM events.

debug ip pim

- Use to show information on the selected event.
- To control the type of events displayed, specify a severity level.
- To control how much information to display, specify a verbosity level.
- Example

```
host1#debug ip pim events severity 1 verbosity low
```
- Use the **no** version to disable the display.

undebug ip pim

- Use to turn off the display of information previously enabled with the **debug ip pim** command.

```
host1#undebug ip pim events
```
- There is no **no** version.

Monitoring PIM Settings

You can use the **show ip pim** commands to display information about PIM settings.

show ip pim auto-rp

- Use to display information about RP routers and the RP mapping agent in a PIM SM environment.
- Field descriptions
 - › Configured with ttl – number of hops for which the RP discovery message is valid
 - › Using interface addr – IP address of the interface from which the system sends RP discovery messages
 - › Interval – time interval at which the system sends RP discovery messages
 - › PIM AutoRP candidate RP mapping(s) – routers that the RP mapping agent is evaluating to determine an RP router for this interface

- Example 1

```
host1:1#show ip pim auto-rp
This PIM router is an Auto RP mapping agent.
Configured with ttl 64
[ Using interface addr 121.0.0.1, interval 60 ].
PIM AutoRP candidate RP mapping(s)
```

- Example 2

```
host1:1#show ip pim auto-rp
This PIM router is _not_ an Auto RP mapping agent.
PIM AutoRP candidate RP mapping(s)
Candidate RP 122.0.0.1
  Group(s) 224.0.0.0/4, AutoRP, ttl 64, interval 60, from access List 1
Candidate RP 122.0.0.1
  Group(s) 224.0.1.39/32 (negative), AutoRP, ttl 64, interval 60, from access List 1
Candidate RP 122.0.0.1
  Group(s) 224.0.1.40/32 (negative), AutoRP, ttl 64, interval 60, from access List 1
```

show ip pim dense-mode sg-state

- Use to display information for each (Source, Group) entry for PIM DM.
- Field descriptions
 - › (Source, Group) pair – IP addresses of multicast source and group
 - › EntryExpires – time until the (Source, Group) pair entry expires
 - › RPF Route – reverse path forwarding route
 - › IIF – IP address of incoming interface
 - › UpNbr – IP address of upstream neighbor
 - › Pruned Oifs – Outgoing interfaces that have been pruned
 - Address – IP address of outgoing interface
 - IfId – index of the interface
 - Pruned due to – reason for prune: assert or explicit prune
 - Pruned time remaining – time in seconds until the prune expires
- Example

```
host1:8#show ip pim dense-mode sg-state
PIM DM route table and pruned oif information
<122.0.0.1, 224.0.1.39>  EntryExpires: 99
RPF Route: 122.0.0.0/255.0.0.0  IIF: 107.0.8.4  UpNbr: 107.0.4.8
Pruned Oifs:
Address: 108.0.8.5  IfId: 95
Pruned due to assert
Pruned time remaining 129
```

```
<130.0.0.2, 224.0.1.39>  EntryExpires: 100
RPF Route: 130.0.0.0/255.0.0.0  IIF: 107.0.8.4  UpNbr: 107.0.4.8
Pruned Oifs:
  Address: 108.0.8.5  IfId: 95
  Pruned due to assert
  Pruned time remaining 130
<121.0.0.1, 224.0.1.40>  EntryExpires: 102
RPF Route: 121.0.0.0/255.0.0.0  IIF: 107.0.8.4  UpNbr: 107.0.4.8
Pruned Oifs:
  Address: 108.0.8.5  IfId: 95
  Pruned due to assert
  Pruned time remaining 133
```

show ip pim interface

- Use to display information about PIM interfaces.
- Provide an interface type and specifier (such as atm 3/0) to display information about that interface only. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Specify the **count** option to view the number of multicast packets that the interface has sent and received.
- Field descriptions
 - › Interface Addr – IP address of the interface
 - › Interface Name – type and identifier of the interface. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Ver – version of PIM running on this interface
 - › Mode – PIM mode running on this interface: sparse, dense, or sparse-dense
 - › Nbr Count – number of neighbors connected to this interface
 - › Hello Intvl – time interval at which the interface sends hello messages to neighbors
 - › DR Address – address of the DR
- Example

```
host1#show ip pim interface
```

```
PIM Interface Table
```

Interface Addr	Interface Name	Ver	Mode	Nbr Count	Hello Intvl	DR Addr
108.0.8.5	atm2/1.108	2	SparseDense	1	30	108.0.8.5
107.0.8.4	atm2/1.109	2	SparseDense	1	30	107.0.8.4
111.0.8.9	atm2/0.110	2	SparseDense	1	30	111.0.9.8
110.0.8.12	loopback8	2	SparseDense	0	30	110.0.8.12

show ip pim neighbor

- Use to display information about PIM neighbors that the system discovered.
- Provide an interface type and specifier (such as atm 3/0) to display information about that interface only. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Field descriptions
 - › Neighbor Addr – IP address of the neighbor
 - › Interface Name – type and specifier of the interface to which the neighbor connects. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Uptime – time since the router discovered this neighbor
 - › Expires – time available for the neighbor to send a hello message to the interface. If the neighbor does not send a hello message during this time, it will no longer be a neighbor.
 - › Ver – version of PIM that the neighbor is running
 - › Mode – PIM mode that the neighbor is using: dense, sparse, or sparse-dense
- Example

```
host1#show ip pim neighbor
```

```
PIM Neighbor Table
```

Neighbor Addr	Interface Name	Uptime	Expires	Ver	Mode
107.0.4.8	atm2/1.109	1d15:47:35	00:01:41	2	SparseDense
108.0.5.8	atm2/1.108	1d15:47:34	00:01:42	2	SparseDense
111.0.9.8	atm2/0.110	1d15:48:02	00:01:44	2	SparseDense

show ip pim remote-neighbor

- Use to view information about PIM remote neighbors.
- Field descriptions
 - › Remote Nbr – IP address of remote neighbor
 - › OurEndAddr – IP address of local interface, such as the local endpoint of a tunnel, that transmits data to remote neighbor
 - › Ver – version of PIM running on the local interface
 - › Mode – PIM mode running on the local interface; always PIM sparse mode
 - › Nbr Count – number of remote neighbors detected: 0 or 1
 - › Hello Intvl – time interval at which the interface sends hello messages to neighbors
 - › DR Addr – address of DR
 - › In interface – type and identifier of the interface on which PIM router receives packets from remote neighbor. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Out interface – type and identifier of the interface on which PIM router sends packets to remote neighbor. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.

```

host1:boston#show ip pim remote-neighbor
PIM RemoteNbr Table
RemoteNbr Addr   OurEnd Addr       Ver Mode           Nbr   Hello   DR Addr
Count Intvl
10.2.23.45      172.16.78.3      2   Sparse           1     30     192.168.3.41
  In interface : atm2/1.109
  Out interface: atm2/1.108

```

show ip pim rp

- Use to display information about PIM group-to-RP mappings.
- Specify the address of a group to view PIM group-to-RP mappings for a particular group.
- To display all RP-to-group mappings that the system has recorded, specify the **mapping** keyword.
- Field descriptions
 - › Group – prefix of the multicast group
 - › RP – IP address of RP router for the multicast group
 - › priority – this field is not functional
 - › via Auto RP/static RP – method by which the RP router was assigned
 - › expiryTime – time in seconds at which the RP mapping becomes invalid, unless the mapping agent reassigns the RP router to this group
- Example

```

host1:8#show ip pim rp mapping
PIM Group-to-RP mapping(s)
Group(s) 224.0.0.0/4
  RP 122.0.0.1, priority 0, via AutoRP, expiryTime 88
Group(s) 224.0.1.39/32 (negative)
  RP 122.0.0.1, priority 0, via AutoRP (Negative), expiryTime 88
Group(s) 224.0.1.40/32 (negative)
  RP 122.0.0.1, priority 0, via AutoRP (Negative), expiryTime 88

```

show ip pim rp-hash

- Shows which RP router a multicast group is using.
- Field descriptions
 - › Group – multicast group
 - › RP – RP router for the multicast group
 - › priority – this field is not functional
 - › via Auto RP/static RP – method by which the RP router was assigned
 - › expiryTime – time in seconds at which the RP mapping becomes invalid, unless it is renewed by the mapping agent
- Example

```

host1:2#show ip pim rp-hash 232.1.1.1
Group(s) 224.0.0.0/4
  RP 122.0.0.1, priority 0, via AutoRP, expiryTime 128

```

show ip pim sparse-mode sg-state

- Use to display information for each (Source, Group) entry for PIM SM.
- Field descriptions
 - › Group-to-RP mapping – IP addresses and network mask of multicast group
 - › RP – IP address of RP router
 - › RPF route – IP address and network mask of RPF route
 - › IIF – IP address of the incoming interface for RPF route
 - › UpNbr – IP address of upstream neighbor
 - › Oifs – outgoing interface
 - › Auto RP Discovery SELF oif – indicates that RP router for this group was assigned via auto-RP
 - › Address – IP address of outgoing interface
 - › Interface – type and specifier of the interface. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Joined as – type of mapping
 - (S,G) – mapping from a specific source to a specific group
 - (*,G) – mapping from any source to a specific group
 - (*,*,RP) – mapping from any source to any group
 - › Count of entries – total count of (Source, Group) pair mappings
- Example

```

host1:2#show ip pim sparse-mode sg-state
PIM SM route table and oif information
<*, 224.0.1.40>
  Group-to-RP mapping: 224.0.0.0/240.0.0.0   RP: 123.0.0.1
  RPF Route: 123.0.0.0/255.0.0.0   IIF: 106.0.7.3   UpNbr:
106.0.3.7
  Oifs:
    Auto RP Discovery SELF oif.
    Joined as <*, G>

<*, 225.1.2.3>
  Group-to-RP mapping: 224.0.0.0/240.0.0.0   RP: 123.0.0.1
  RPF Route: 123.0.0.0/255.0.0.0   IIF: 106.0.7.3   UpNbr:
106.0.3.7
  Oifs:
    Address: 78.7.7.7   Interface: loopback7
    Local group membership present.

<*, 232.1.1.1>
  Group-to-RP mapping: 224.0.0.0/240.0.0.0   RP: 123.0.0.1
  RPF Route: 123.0.0.0/255.0.0.0   IIF: 106.0.7.3   UpNbr:
106.0.3.7
  Oifs:
    Address: 78.7.7.7   Interface: loopback7
    Local group membership present.

```

```

<10.0.1.8, 232.1.1.1>      EntryExpires: 143
  Group-to-RP mapping: 224.0.0.0/240.0.0.0  RP: 123.0.0.1
  RPF Route: 10.0.0.0/255.0.0.0  IIF: 106.0.7.3  UpNbr:
  106.0.3.7
  Oifs:
    Address: 78.7.7.7  Interface: loopback7
    Joined as <*, G>

Count of entries - <S, G>      : 1
                   <*, G>      : 3
                   <*, *, RP>: 0

```

show ip pim sparse-mode unicast-route

- Use to display the unicast routes that PIM SM is using.
- Field descriptions
 - › Route – IP address and network mask for the unicast route
 - › RpfNbr – RPF neighbor
 - › lif – incoming interface for the unicast route
 - › Pref – preference for the unicast route
 - › Metric – value of metric for the unicast route (type of metric varies with the unicast protocol)
 - › Count of entries – number of unicast routes that PIM SM is using.
- Example

```

host1:2#show ip pim sparse-mode unicast-route
PIM SM unicast route table information
Route                               RpfNbr                               Iif                               Pref  Metric
-----
122.0.0.0 /255.0.0.0                122.0.0.1                            255   1
Count of entries: 1

```

show ip pim spt-threshold

- Use to display the threshold for switching to the shortest path tree at a PIM DR.
- Field descriptions
 - › Access List Name – name of the IP access list that specifies the groups to which the threshold applies
 - › SptThreshold (in kbps) – value at which PIM SM should switch from a shared tree to an SPT. A value of infinity indicates that PIM SM should never switch to an SPT.

```

host1:2#show ip pim spt-threshold
Access List Name                    SptThreshold(in kbps)
-----
1                                    infinity

```

DVMRP

The system supports Distance Vector Multicast Routing Protocol (DVMRP) on VRs to forward multicast datagrams through a network. DVMRP is an interior gateway protocol that supports operations within an autonomous system, but not between autonomous systems. The multicast backbone of the Internet, MBONE, uses DVMRP to forward multicast datagrams.

DVMRP is a dense-mode multicasting protocol and therefore uses a broadcast and prune mechanism. The protocol builds an SRT in a similar way to PIM DM (see Figure 3-3). DVMRP routers flood datagrams to all interfaces except the one that provides the shortest unicast route to the source. DVMRP uses pruning to prevent unnecessary sending of multicast messages through the SRT.

A DVMRP router sends prune messages to its neighbors if it discovers that:

- The network to which a host is attached has no active members of the multicast group.
- All neighbors, except the next-hop neighbor connected to the source, have pruned the source and the group.

When a neighbor receives a prune message from a DVMRP router, it removes that neighbor from its (Source, Group) pair table, which provides information to the multicast forwarding table.

If a host on a previously pruned branch wants to join a multicast group, it sends an IGMP message to its first-hop router. The first-hop router then sends a graft message upstream.

Identifying Neighbors

In this implementation of DVMRP, a *neighbor* is a directly connected DVMRP router. When you enable DVMRP on an interface, the associated VR adds information about local networks to its DVMRP routing table. The VR then sends probe messages periodically to learn about neighbors on each of its interfaces. To ensure compatibility with other DVMRP routers that do not send probe messages, the VR also updates its DVMRP routing table when it receives route report messages from such routers.

Advertising Routes

As its name suggests, DVMRP uses a distance vector routing algorithm. Such algorithms require that each router periodically inform its neighbors of its routing table. DVMRP routers advertise routes by sending DVMRP report messages. For each network path, the receiving router picks the neighbor advertising the lowest cost and adds that entry to its routing table for future advertisement.

The cost or metric for this routing protocol is the hop count back to the source. The hop count for a network device is the number of routers on the route between the source and that network device.

Table 3-2 shows an example of the routing table for a DVMRP router.

Table 3-2 Sample routing table for a DVMRP router

Source Subnet	Subnet Mask	From Router	Metric	Time Before Entry Is Deleted from Routing Table	Input Port	Output Port
143.2.0.0	255.255.0.0	143.32.44.12	4	85	3/0	4/0, 4/1
143.3.0.0	255.255.0.0	143.2.55.23	2	80	3/1	4/0, 4/1
143.4.0.0	255.255.0.0	143.78.6.43	3	120	3/1	4/0, 4/1

The DVMRP router maintains a (Source, Group) pair table that provides information to the multicast forwarding table. The (Source, Group) pair table is based on:

- Information from the DVMRP routing table
- Information learned from prune messages
- If IGMP and DVMRP are on the same interface, group information learned from IGMP

The (Source, Group) pair table includes a route from each subnetwork that contains a source to each multicast group of which that source is a member. These routes can be static or learned routes. Table 3-3 shows an example of the (Source, Group) pair table for DVMRP.

Table 3-3 Example of DVMRP (Source, Group) pair table

Source Subnet	Multicast Group	Time Before Entry Is Deleted from Routing Table	Input Port	Output Port
143.2.0.0	230.1.2.3	85	3/0	4/0, 4/1
	230.2.3.4	75	3/0	4/0, 4/1
	230.3.4.5	60	3/0	4/1

Table 3-3 Example of DVMRP (Source, Group) pair table (continued)

Source Subnet	Multicast Group	Time Before Entry Is Deleted from Routing Table	Input Port	Output Port
	230.4.5.6	90	*	4/0
143.3.0.0	230.1.2.3	80	3/1	4/0, 4/1

*. No value for the input port indicates that the interface is associated with a protocol other than DVMRP.

Enabling DVMRP on a VR

By default, DVMRP is enabled on the system. To enable DVMRP on a VR:

- 1 Enable multicast routing.
`host1(config)#ip multicast-routing`
- 2 (Optional) Create a VR or access a VR context.
`host1(config)#virtual-router boston`



Note: If you do not specify a VR, you can configure DVMRP on the default router.

You must now enable and configure DVMRP on one or more interfaces. See *Activating DVMRP on an Interface*. You can also set DVMRP limits for the VR. See *Configuring DVMRP Limits*.

Activating DVMRP on an Interface

By default, DVMRP is not activated on an interface. Configuring any DVMRP parameter on an interface automatically activates DVMRP on that interface. You can also activate DVMRP on an interface and use the default parameters.

ip dvmrp

- Use to activate DVMRP on an interface.
- This command automatically creates and enables DVMRP processing on the current VR.
- Issuing this command identifies this interface as one that DVMRP owns.
- Example
`host1:boston(config-if)#ip dvmrp`
- Use the **no** version to remove DVMRP from an interface.

Configuring DVMRP Limits

You can configure DVMRP and IGMP on the same interface. If you configure IGMP and DVMRP on an interface, the system considers that DVMRP owns the interface.



Note: You cannot configure DVMRP and PIM on the same interface.

When you have enabled DVMRP processing on a VR, you can configure the following settings for that VR:

- The number of routes that the VR advertises on each interface.
- A maximum number of DVMRP routes at which the system generates a system log warning message and an SNMP trap.

ip dvmrp route-hog-notification

- Use to set the number of DVMRP routes that the system can record before it generates a system log warning message.
- The warning allows you to identify routers that are injecting large numbers of routes into the MBONE.
- Example

```
host1:boston(config)#ip dvmrp route-hog-notification 5000
```

- Use the **no** version to revert to the default setting, 10,000 routes.

ip dvmrp route-limit

- Use to limit the number of routes that the system can advertise on each interface.
- Example

```
host1:boston(config)#ip dvmrp route-limit 5000
```

- Use the **no** version to restore the default, 7000 routes.

Filtering DVMRP Reports

You can configure an interface to accept only reports with routes that appear on a standard IP access list. You can refine the set of accepted routes further, by defining a second access list of neighbors who can supply the specified routes.

For example, suppose you define an access list that specifies that the router accepts only reports for the route 172.16.2.0/24. You then define a second access list that specifies that only neighbors 192.168.1.1 and 193.168.1.1 can supply this route. If neighbor 192.168.2.2 supplies the route, the DVMRP router rejects this report.

You can also modify the value (distance) that the router associates with a DVMRP route when it computes the RPF interface for the source of a multicast packet. By default, the router associates a distance of 0 with DVMRP routes; this value indicates that the router should use DVMRP, rather than a unicast routing protocol, to transport multicast datagrams.

However, in a configuration where PIM discovers multicast routes and a unicast routing protocol performs RPF lookups, you can increase the administrative distance to favor the unicast protocol.

For information about defining access lists, see *Chapter 1, Configuring Routing Policy*.

ip dvmrp accept-filter

- Use to filter routes in DVMRP reports in accordance with a standard IP access list.
- Specify a standard IP access list of sources for which the interface will accept routes.
- To favor a unicast routing protocol, specify a DVMRP administrative distance.
- To restrict the neighbors from whom reports for routes on the first list will be accepted, specify a neighbor list.
- Example

```
host1:boston(config-if)#ip dvmrp accept-filter boston-list 4  
neighbor-list boston-neighbors
```

- Use the **no** version to disable a filter.

Configuring DVMRP Summary Addresses

You can configure an interface to advertise a summary address with a known metric rather than a more specific route. DVMRP advertises the summary address if the DVMRP routing table contains a more specific route that matches the address and mask of the summary address.

If you want to advertise all routes rather than a summary, disable automatic summarization on the interface. By default, the system automatically summarizes DVMRP routes. DVMRP automatic summarization maps a unicast subnet route to a classful network number route when the subnet has a different network number from the IP address of the interface (or tunnel) over which the advertisement travels. If the interface is unnumbered, the system compares the network number of the numbered interface to the IP address to which the unnumbered interface points.

If you configure a summary address on an interface and do not disable automatic summarization, the interface advertises the least specific address.

ip dvmrp auto-summary

- Use to reenable the system to summarize routes automatically for this interface. By default, automatic summarization is enabled.
- Example

```
host1:boston(config-if)#ip dvmrp auto-summary
```
- Use the **no** version to disable automatic summarization for this interface.

ip dvmrp summary-address

- Use to advertise DVMRP summary addresses on an interface. By default, an interface advertises only summary addresses generated by automatic summarization.
- If you configure multiple overlapping summary addresses on an interface, the one with the shortest mask takes preference.
- The default metric value is 1.
- Example

```
host1:boston(config-if)#ip dvmrp summary-address 192.48.1.2
255.255.255.0 metric 1
```
- Use the **no** version to stop advertising a summary address on the interface.

Changing the Metric for a Route

The metric for DVMRP is hop count. For example, a route with two hops over a slow serial line is preferable to a route with three hops over a faster optical line.

The system increments DVMRP routes in incoming reports by a default metric of one and in outgoing reports by a default of 0. You can change the metric for an interface to promote or demote the preference for associated routes.

ip dvmrp metric-offset

- Use to adjust the number of hops associated with a route. This action specifies that the route is more efficient or less efficient than an alternative route.
- Use the **in** keyword to specify the number of hops by which the system increments a DVMRP route advertised in incoming DVMRP reports. This option is the default.
- Use the **out** keyword to specify the number of hops by which the system increments a DVMRP route advertised in outgoing DVMRP reports.
- Example

```
host1:boston(config-if)#ip dvmrp metric-offset in 3
```
- Use the **no** version to revert to the default settings: 1 for incoming reports and 0 for outgoing reports.

Importing Routes from Other Protocols

You can import routing information from other protocols into the DVMRP routing table. To do so:

- 1 If you want to use IS-IS, OSPF, or RIP routes, make those routes available to multicasting protocols. See *Using Unicast Routes for RPF*, earlier in this chapter.

```
host1(config)#router ospf
host1(config-router)#ip route-type multicast
```

- 2 Access Router Configuration mode for DVMRP.

```
host1:boston(config)#router dvmrp
```

- 3 Specify a route map.

```
host1:boston(config-router)#route-map boston-map atm 3/2
```

- 4 Import information from one type of routing domain into another.

```
host1:boston(config-router)#redistribute bgp 100 route-map
boston-map
```

redistribute

- Use to import information from one type of routing domain to another.
- Specify the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **isis**, **ospf**, **static [ip]**, and **connected**. The keyword **static [ip]** is used to redistribute IP static routes.
- Use the **static ip** keyword to redistribute static IP routes into DVMRP.
- Use the keyword **connected** to redistribute routes that are established automatically because IP is enabled on an interface.
- Use the **route-map** keyword to interrogate the route map to filter imported routes from the source routing protocol to the current routing protocol. If you do not specify the route-map option, all routes are redistributed. If you specify the route-map option, but no route map tags are listed, no routes will be imported.

- Example: Importing routing information from BGP into DVMRP

```
host1:boston(config-router)#redistribute bgp 100 route-map
boston-map
```

- Use the **no** version of this command to disable redistribution.

-

- Use to specify a route map.

```
host1:boston(config-router)#route-map boston-map atm 3/2
```

- Use the **no** version to delete the route map. If you do not specify an interface, it removes the global route map if one exists.

router dvmrp

- Use to create and enable DVMRP processing on a VR or to access DVMRP Router Configuration mode.
- Example

```
host1:boston(config)#router dvmrp
```
- Use the **no** version to remove DVMRP from the VR.

Specifying Routes to Be Advertised

By default, if DVMRP owns an interface, that interface advertises all DVMRP routes it has learned to its neighbors. You can specify the routes that the interface advertises by issuing the **ip dvmrp announce-filter** command in conjunction with a standard IP access list. The IP access list defines the DVMRP routes that will be advertised.

ip dvmrp announce-filter

- Use to specify the DVMRP routes that an interface will advertise.
- Specify a standard IP access list of routes that the interface will advertise.
- Example

```
host1:boston(config-if)#ip dvmrp announce-filter boston-list
```
- Use the **no** version to allow the interface to advertise all DVMRP routes that it has learned.

Preventing Dynamic Route Distribution

By default, if you make changes to a route map, the system dynamically redistributes the routes in DVMRP. To prevent this dynamic redistribution, use the **disable-dynamic-redistribute** command.

disable-dynamic-redistribute

- Use to halt the dynamic redistribution of routes that are initiated by changes to a route map.
- Dynamic redistribution is enabled by default.
- Example

```
host1(config-router)#disable-dynamic-redistribute
```
- There is no **no** version.

Exchanging DVMRP Unicast Routes

DVMRP maintains its own unicast routing table, based on distance vector calculations. The routing table defines the best-known distance to each destination and how to get there. The router updates the tables by

exchanging information with its neighbors. The DVMRP routing table is used solely for RPF lookups.

By default, if DVMRP owns an interface, that interface exchanges DVMRP unicast routes with its neighbors, and you cannot disable the exchange of routes. However, you can enable and disable the exchange of DVMRP unicast routes on interfaces that DVMRP does not own.

When an interface exchanges DVMRP routes, the router obtains routes from DVMRP report messages and stores them in its DVMRP routing table. Other multicast protocols, such as PIM, can then use these routes for RPF lookups. With this feature, PIM can use the DVMRP routing table even when the router is not running DVMRP.

All interfaces, including tunnels, support DVMRP unicast routing. DVMRP tunnels use DVMRP multicast routing to support DVMRP unicast routing.

ip dvmrp unicast-routing

- Use to enable the exchange of DVMRP unicast routes on an interface not owned by DVMRP.
- Example

```
host1:boston(config-if)#ip dvmrp unicast-routing
```
- Use the **no** version to disable the exchange of DVMRP unicast routes on an interface not owned by DVMRP.

Disabling and Removing DVMRP

You can disable DVMRP on a VR or an interface without removing the configuration. You can also remove DVMRP from a VR or an interface.

disable

- Use to disable DVMRP processing on a VR without removing the DVMRP configuration. By default, DVMRP processing is enabled.
- Example

```
host1:boston(config-router)#disable
```
- Use the **no** version to reenables DVMRP processing on a VR.

ip dvmrp disable

- Use to disable DVMRP processing on an interface without removing the DVMRP configuration.
- Example

```
host1:boston(config-if)#ip dvmrp disable
```
- Use the **no** version to reenables DVMRP on an interface.

ip dvmrp

- Use to activate DVMRP on an interface.
- This command automatically creates and enables DVMRP processing on the current VR.
- Issuing this command identifies this interface as one that DVMRP owns.
- Example

```
host1:boston(config-if)#ip dvmrp
```
- Use the **no** version to remove DVMRP from an interface.

router dvmrp

- Use to create and enable DVMRP processing on a VR or to access Router Configuration mode for DVMRP.
- Example

```
host1:boston(config)#router dvmrp
```
- Use the **no** version to remove DVMRP from the VR.

Deleting DVMRP Routes

You can clear one or more routes from the DVMRP routing table. However, if you do so, the routes may reappear in the routing table if they are rediscovered.

clear ip dvmrp routes

- Use to delete DVMRP routes from the routing table.
- If you do not specify any options, the system removes all routes except those associated with its own interfaces from the DVMRP table.
- If you specify an IP address but not a subnet mask, the system removes the longest route to that IP address from the DVMRP table.
- If you specify a subnet mask, the system removes that specific route from the DVMRP table.
- Example

```
host1:boston#clear ip dvmrp routes
```
- There is no **no** version.

Configuring DVMRP Tunnels

DVMRP tunnels allow the exchange of IP multicast traffic between routers separated by networks that do not support multicast routing. For information about DVMRP tunnels, see *Chapter 4, Configuring IP Tunnels*.

Monitoring DVMRP

You can establish a reference point for DVMRP statistics by setting the statistics counters to zero.

You can display DVMRP information with the **show ip dvmrp** commands.

baseline ip dvmrp

- Use to set the counters for IGMP statistics to zero.
- Example

```
(host1)#baseline ip dvmrp
```
- There is no **no** version.

show ip dvmrp

- Use to display DVMRP information for a VR.
- Field descriptions
 - › Dvmrp Admin State – state of DVMRP in the software: enabled or disabled
 - › Mcast Admin State – state of multicasting in the software: enabled or disabled
 - › Dvmrp Version – version of DVMRP with which this software is compatible
 - › GenerationID – a number the router generates each time it reboots; when the number changes, neighbors discard all information previously learned from the router
 - › NumRoutes – number of routes in the DVMRP routing table
 - › NumTrigdRts – number of routes waiting to be advertised, because a parameter for the route changed
 - › ReachableRoutes – number of routes that the router can currently reach
 - › RouteHogNotification – number of DVMRP routes that the system can record before it generates a system log warning message
 - › RouteLimit – maximum number of routes that the system can advertise on each interface
 - › Send-S32-Prunes-Only – indicates whether or not the router sends only S-32 prunes
 - True – router sends only S-32 prunes and grafts to ensure compatibility with other protocols, such as PIM
 - False – router sends S-32 and S/Prefix grafts and prunes

- Example

```
host1:boston>show ip dvmrp
Routing Process DVMRP - Distance Vector Multicast Routing
Protocol
  Dvmrp Admin State:           Enabled
  Mcast Admin State:          Enabled
  Dvmrp Version:               3.255
  GenerationID:                0x39aa07d3
  NumRoutes:                   7
  NumTrigdRts:                 0
  ReachableRoutes:             7
  RouteHogNotification:        10000
  RouteLimit:                  7000
  Send-S32-Prunes-Only:        false
```

show ip dvmrp interface

- Use to display DVMRP parameters for the specified interfaces.
- Field descriptions
 - › Interface – type and identifier of the interface connected to a source. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › SourceAddress – IP address of the interface or, for an unnumbered interface, the address of the loopback interface
 - › Network/Mask – network and mask of the subnet on which the interface resides
 - › Received Bad Packets/RBdPk – number of bad packets received on this interface
 - › Received Bad Routes/RBdRt – number of bad routes received on this interface
 - › Routes Sent/SntRt – number of bad routes advertised on this interface
 - › Administrative State – configured state of DVMRP on this interface: enabled or disabled
 - › Summary Address – specific summary address that this interface should advertise
 - › auto-summary – status of automatic summarization: enabled or disabled
 - › metric-offset in – number of hops by which the system increments a DVMRP route advertised in incoming DVMRP reports
 - › metric-offset out – number of hops by which the system increments a DVMRP route advertised in outgoing DVMRP reports
 - › accept-filter(s) – names of IP access lists that specify the sources for which the interface accepts routes.

- Example 1

```

host1:v3#show ip dvmrp interface
Interface: atm5/0.14
  SourceAddress:                14.0.1.1
  Network/Mask:                 14.0.1.1/8
  Received Bad Packets:         0
  Received Bad Routes:         0
  Routes Sent:                 2
  Administrative State:        Enabled
  Summary Address(es)
    None Configured
  auto-summary:                Enabled
  metric-offset in:            1
  metric-offset out:          0
  accept-filter(s):           None Configured

```

- Example 2

```

host1:boston#show ip dvmrp interface brief
Interface      SourceAddress      Network/Mask      RbDpK RbDRt SntRt
atm5/0.14      14.0.1.1           14.0.1.1/8       0     0     2
atm5/0.15      15.0.1.1           15.0.1.1/8       0     0     2

```

show ip dvmrp mroute

- Use to display information about DVMRP routes to multicast groups.
- Field descriptions
 - › For each (Source, Group) pair:
 - No Upstream Prune – router has sent no prune messages for this group
 - Uptime – time, in seconds, that this (Source, Group) pair entry has been in the routing table
 - RPF Interface – interface that provides the shortest path back to the source
 - Outgoing interface list – types and identifiers of interfaces through which the VR forwards DVMRP messages, such as atm3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Example

```

host1:boston#show ip dvmrp mroute
IP DVMRP Multicast Routing Table
(40.0.0.0/16, 228.1.1.1) Uptime: 77
  Upstream Prune: none
  RPF Interface
    atm5/0.40
  Outgoing interface list:
    atm5/0.31

```

show ip dvmrp neighbor

- Use to display information about DVMRP neighbors.
- Field descriptions
 - › Interface – interface type and identifier, such as atm3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
 - › Neighbor Address/NbrAddress – IP address of the neighbor
 - › Neighbor upTime/UpTime – length of time, in seconds, that this router has been a neighbor
 - › Neighbor Major Version/Maj – major number of the DVMRP version on the neighbor
 - › Neighbor Minor version/Min – minor number of the DVMRP version on the neighbor
 - › Neighbor capabilities/Cap – capability of the neighbor
 - Prune/P – can send prune messages
 - GenerationId/G – can create a GenID number
 - Mtrace/M – can trace multicast routes
 - Netmask/N – can send prunes and grafts with a network mask address
 - › Neighbor State/State – status of the communications with the neighbor
 - Active – router is able to communicate with this neighbor
 - Down – neighbor is down
 - Ignoring – router is not accepting message from this neighbor
 - Oneway – router is receiving messages from the neighbor, but the neighbor does not include the router's IP address in the messages. This state can indicate a starting transition, or a problem.
 - › Generation ID – number that the neighbor generates each time it boots; when the number changes, the VR discards all information previously learned from the router.
 - › Routes Received – number of routes learned from this neighbor
 - › Bad Routes Received – number of bad routes received from this neighbor
 - › Bad Packets Received – number of bad packets received from this neighbor
- Example 1

```
host1:boston#show ip dvmrp neighbor
Neighbor Address:          14.0.0.1
Interface:                 atm5/0.14
Neighbor upTime:          28
Neighbor Major Version:   3
Neighbor Minor Version:   255
Neighbor Capabilities:    Prune GenerationId Mtrace NetMask
Neighbor State:           Active
Geneneration ID:          0x3a13fbc2
Routes Received:          1
Bad Routes Received       0
Bad Packets Received:     0
```

- Example 2

```
host1:v3#show ip dvmrp neighbor brief
  Interface          NbrAddress      UpTime Maj Min Cap
  State
  atm5/0.14          14.0.0.1        32   3 255 PGMN
  Active
  atm5/0.15          15.0.0.1        34   3 255 PGMN
  Active
```

show ip dvmrp route

- Use to display information about DVMRP routes.
- Specify an IP address to display the best route to that address.
- Specify an IP address and subnet mask to display the route that exactly matches this IP address and subnet mask
- Specify an interface type and specifier to display routes associated with that interface.
- Specify the **brief** keyword to view a summary of information.
- Field descriptions
 - › Prefix – IP address of the network
 - › Length – length of the subnet mask for the network
 - › usNbr/Owner – IP address of the upstream neighbor associated with this route or a description of the origin of the route
 - DVMRP Local – route is associated with a directly attached network
 - DVMRP Aggregate – route is an aggregate route determined by summarization
 - › Metric – metric associated with this interface for this route
 - › ExpireTime – time until the VR starts the process for removing the route
 - › UpTime – length of time the route has been in the DVMRP routing table
 - › Interface – type and identifier for the interface, such as atm3/0. For details about interface types and specifiers, see *ERX Command Reference Guide, About This Guide*.
- Example 1

```
host1:boston>show ip dvmrp route
Prefix/Length      usNbr/Owner      Metric ExpireTime UpTime Interface
14.0.0.0/8         Dvmrp Local      1      Never      18   atm5/0.14
  Downstream Interface(s)
  Interface
  atm5/0.15
15.0.0.0/8         Dvmrp Local      1      Never      18   atm5/0.15
  Downstream Interface(s)
  None
25.0.0.0/8         14.0.0.1         2      129       11   atm5/0.14
  Downstream Interface(s)
  Interface
  atm5/0.15
```

- Example 2

```
host1:v3#show ip dvmrp route brief
```

Prefix/Length	usNbr/Owner	Metric	ExpireTime	UpTime	Interface
14.0.0.0/8	Dvmrp Local	1	Never	26	atm5/0.14
15.0.0.0/8	Dvmrp Local	1	Never	26	atm5/0.15
25.0.0.0/8	14.0.0.1	2	121	19	atm5/0.14

show ip dvmrp routeNextHop

- Use to display information about the next hop.
- Field descriptions
 - › addr – IP address of the next-hop router
 - › mlen – mask length of the next-hop router
 - › ifIndex – SNMP ifIndex for the interface that connects to the next hop
 - › Type – description of the next-hop router
 - leaf – neighbor with no downstream neighbors
 - branch – neighbor with downstream neighbors
- Example

```
host1:boston>show ip dvmrp routeNextHop
```

addr/mlen	ifIndex	Type
172.16.0.0/16	4	leaf
172.17.0.0/16	4	leaf
172.18.0.0/16	3	leaf
172.19.0.0/16	3	leaf
172.19.0.0/16	4	branch

BGP Multicasting

BGP multicasting (MBGP) is an extension of the BGP unicast routing protocol. Many of the functions available for BGP unicast are also available for MBGP.

The MBGP extensions specify that BGP can exchange information within different types of *address families*. The address families available are unicast IPv4, multicast IPv4, and VPN-IPv4. When you enable BGP, the system employs unicast IPv4 addresses by default.

You should be thoroughly familiar with BGP before configuring MBGP. See *ERX Routing Protocols Configuration Guide, Vol. 2, Chapter 1, Configuring BGP Routing*, for detailed information on BGP and MBGP.

Investigating Multicast Routes

You can use the **mtrace** command to trace the path that multicast packets take from a source to a destination via a multicast group address. This command is similar to the **tracert** command for investigating unicast routes.

mtrace

- Use this command to trace the path that multicast packets take to a destination.
- Specify the unicast IP address of the source for the packets.
- To direct the packets to a particular destination, specify the unicast address for that destination. If you do not specify a destination, the system traces the route from the device on which you issue the command.
- To direct the packets via a particular multicast group address, specify that multicast group address. If you do not specify a multicast group address, the system traces the route via the Mbone audio multicast group.
- To send the trace to a particular device, specify the IP address of that device. If you do not specify a response address, the system sends the trace to an IP address on the router.
- To investigate a problem at a particular point in the route, specify the maximum number of hops for the trace.
- The trace starts at the destination and works back to the source. The default number of hops is 64.
- Field descriptions
 - › Tracing multicast route from a.a.a.a to b.b.b.b for group c.c.c.c using response address d.d.d.d – a description of the trace as follows:
 - a.a.a.a – IP address of the source
 - b.b.b.b – IP address of the destination
 - c.c.c.c – IP address of the multicast group
 - d.d.d.d – IP address on the router to which the system will send the trace
 - › Received mtrace response packet of length *n* – length of the response packet in bytes
 - › Each line of the trace has the following format: hops ip-address protocol threshold
 - hops – number of hops from the destination to this intermediate router
 - ip-address – IP address of the intermediate router
 - protocol – multicast protocol running on the intermediate router. A value of 12 indicates IGMP; other values comply with *A "traceroute" Facility for IP Multicast – draft-ietf-idmr-traceroute-ipm-07.txt* (July 2000).
 - FwdingCode – forwarding information or error associated with this hop. For example, RPF iif indicates that the request arrived on the expected RPF interface for this source group. For more information about the forwarding information or error codes, see *A "traceroute" Facility for IP Multicast – draft-ietf-idmr-traceroute-ipm-07.txt* (July 2000).

- Example

```
host1#mtrace 100.4.4.4 40.1.1.1 232.1.1.1
```

```
Tracing multicast route from 100.4.4.4 to 40.1.1.1 for group  
232.1.1.1 using response address 10.6.129.56
```

```
(Press ^c to stop.)
```

```
Received mtrace response packet of length 88
```

1. 40.1.1.1 Protocol: PIM(3) FwdingCode: RPF iif(9)
2. 21.2.2.2 Protocol: PIM(3) FwdingCode: Reached RP(8)

- There is no **no** version.

