- Multicast -

Types of "packets"

Three types of packets can exist on an IPv4 network:

Unicast – A packet sent from one host to only one other host. A hub will forward a unicast out all ports. If a switch has a table entry for the unicast's MAC address, it will forward it out only the appropriate port.

Broadcast – A packet sent from one host to *all* hosts on the IP subnet. Both hubs and switches will forward a broadcast out all ports. By definition, a router will not forward a broadcast from one segment to another.

Multicast – A packet sent from one host to a specific group of hosts. Switches, *by default*, will forward a multicast out all ports. A router, *by default,* will not forward a multicast from one segment to another.

Multicast Concepts

Remember, a multicast is a packet sent from one computer to a **group** of hosts. A host must **join** a **multicast group** in order to accept a multicast.

Joining a multicast group can be accomplished statically or dynamically.

Multicast traffic is generally sent *from* a multicast server, *to* multicast clients. Very rarely is a multicast packet sent back from a client to the server.

Multicasts are utilized in a wide range of applications, most notably voice or video systems that have one source "serving" out data to a very specific group of clients.

The key to configuring multicast is to ensure *only* the hosts that require the multicast traffic actually receive it.

* * *

Multicast Addressing

IPv4 addresses are separated into several "classes."

Class A: 1.1.1.1 – 127.255.255.255
Class B: 128.0.0.0 – 191.255.255.255
Class C: 192.0.0.0 – 223.255.255.255
Class D: 224.0.0.0 – 239.255.255.255

Class D addresses have been reserved for multicast. Within the Class D address space, several ranges have been reserved for specific purposes:

- **224.0.0.0 – 224.0.0.255 –** Reserved for routing and other network protocols, such as OSPF, RIP, VRRP, etc.
- **224.0.1.0 – 238.255.255.255 –** Reserved for "public" use, can be used publicly on the Internet. Many addresses in this range have been reserved for specific applications
- **239.0.0.0 – 239.255.255.255 –** Reserved for "private" use, and cannot be routed on the Internet.

The following outlines several of the most common multicast addresses reserved for routing protocols:

- 224.0.0.1 – all hosts on this subnet
- 224.0.0.2 – all routers on this subnet
- 224.0.0.5 – all OSPF routers
- 224.0.0.6 – all OSPF Designated routers
- 224.0.0.9 – all RIPv2 routers
- 224.0.0.10 – all IGRP routers
- 224.0.0.12 – DHCP traffic
- 224.0.0.13 – all PIM routers
- 224.0.0.19-21 – ISIS routers
- 224.0.0.22 – IGMP traffic
- 224.0.1.39 – Cisco RP Announce
- 224.0.1.40 – Cisco RP Discovery

Multicast MAC Addresses

Unfortunately, there is no ARP equivalent protocol for multicast addressing. Instead, a reserved range of MAC addresses were created for multicast IPs. All multicast MAC addresses begin with:

0100.5e

Recall that the first six digits of a MAC address identify the vendor code, and the last 6 digits identify the specific host address. To complete the MAC address, the last **23 bits** of the multicast IP address are used.

For example, consider the following multicast IP address and its binary equivalent:

224.65.130.195 = 11100000.01000001.10000010.11000011

Remember that a MAC address is 48 bits long, and that a multicast MAC must begin with *0100.5e*. In binary, that looks like:

00000001.00000000.01011110.0

Add the last 23 bits of the multicast IP address to the MAC, and we get:

00000001.00000000.01011110.01000001.10000010.11000011

That should be exactly 48 bits long. Converting that to Hex format, our full MAC address would be:

0100.5e41.82c3

How did I convert this to Hex? Remember that hexadecimal is Base 16 mathematics. Thus, to represent a single hexadecimal digit in binary, we would need 4 bits ($2^4 = 16$). So, we can break down the above binary MAC address into groups of four bits:

Binary	0000	0001	0000	0000	0101	1110	0100	0001	1000	0010	1100	0011
Decimal	0	1	0	0	5	14	4	1	8	2	12	3
Hex	0	1	0	0	5	e	4	1	8	2	c	3

Hence the MAC address of *0100.5e41.82c3*.

* * *

Multicast MAC Addresses (continued)

Ready for some more math, you binary fiends?

Calculate what the multicast MAC address would be for the following IP addresses:

225.2.100.15 = 11100001.00000010.01100100.00001111
231.130.100.15 = 11100111.10000010.01100100.00001111

Remember that all multicast MACs begin with:

0100.5e = 00000001.00000000.01011110.0

So, add the last 23 digits of each of the above IP addresses to the MAC address, and we get:

225.2.100.15 = 00000001.00000000.01011110.00000010.01100100.00001111
231.130.100.15 = 00000001.00000000.01011110.00000010.01100100.00001111

In Hex, that would be:

225.2.100.15 = 0100.5e02.640f
231.130.100.15 = 0100.5e02.640f

Wait a second…. That's the *exact* same multicast MAC address, right? Double-checking our math, we see that it's perfect.

Believe it or not, each multicast MAC address can match **32 multicast IP addresses**, because we're only taking the last 23 bits of our IP address.

We already know that all multicast IP addresses MUST begin 1110. Looking at the 225.2.100.15 address in binary:

1110***0001.0***0000010.01100100.00001111

That leaves 5 bits in between our starting 1110, and the last 23 bits of our IP. Those 5 bits could be anything, and the multicast MAC address would be the same. Because $2^5 = 32$, there are 32 multicast IP's per multicast MAC.

According to the powers that be, the likelihood of two multicast systems utilizing the same multicast MAC is rare. The worst outcome would be that hosts joined to either multicast system would receive multicasts from both.

* * *

Multicasts and Routing

A router, by default, will drop multicast traffic, unless a **Multicast routing protocol** is utilized. Multicast routing protocols ensure that data sent from a multicast source are received by (and *only* by) its corresponding multicast clients.

Several multicast routing protocols exist, including:

- **Protocol Independent Multicast** (PIM)
- **Multicast OSPF** (MOSPF)
- **Distance Vector Multicast Routing Protocol** (DVMRP)
- **Core-Based Trees** (CBT)

Multicast routing must be enabled globally on a Cisco router or switch, before it can be used:

Switch(config)# *ip multicast-routing*

Multicast Path Forwarding

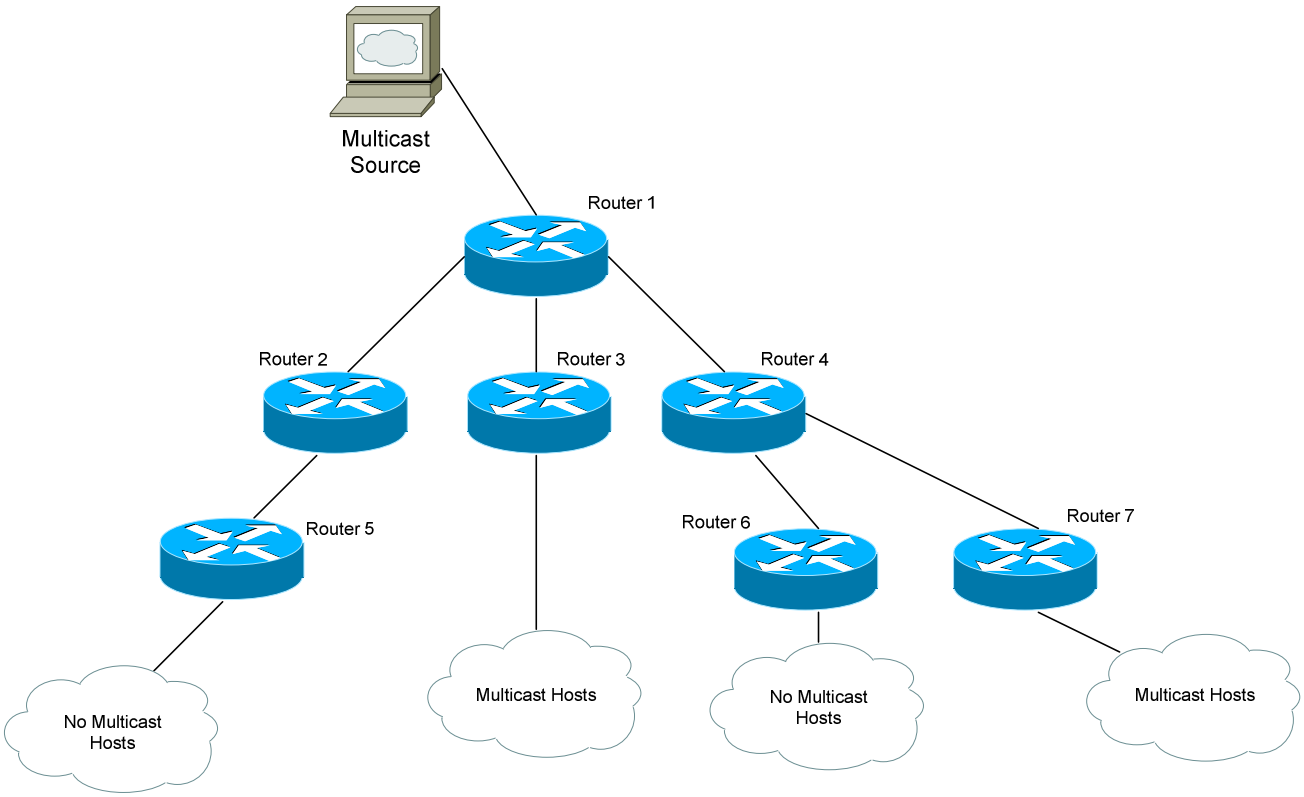
Normally, routers build routing tables that contain **destination** addresses, and route packets *towards* that destination. With multicast, routers are concerned with routing packets *away* from the multicast source. This concept is called **Reverse Path Forwarding (RPF)**.

Multicast routing protocols build tables that contain several elements:

- The multicast **source**, and its associated multicast address (labeled as "**S,G**", or "**Source,Group")**
- **Upstream** interfaces that point *towards* the source
- **Downstream** interfaces that point *away* from the source towards multicast hosts.

* * *

Multicast Path Forwarding Example



A router interface will not be designated as a **downstream** interface unless multicast hosts actually exist downstream. In the above example, no multicast hosts exist downstream of Router 5.

In fact, because no multicast hosts exist downstream of Router 1 towards Router 2, no multicast traffic for this multicast group will be forwarded down that path. Thus, Router 1's interface connecting to Router 2 will not become a downstream port.

This pruning allows for efficient use of bandwidth. No unnecessary traffic is sent down a particular link. This "map" of which segments contain multicast hosts is called the **multicast tree**. The multicast tree is dynamically updated as hosts join or leave the multicast group (otherwise known as **pruning** the branches).

By designating upstream and downstream interfaces, the multicast tree remains loop-free. No multicast traffic should ever be sent back upstream *towards* the multicast source.

* * *

Internet Group Management Protocol (IGMP)

Remember, multicast works by having a **source** send data to a specific set of **clients** that belong to the same multicast **group.** The multicast group is configured (or assigned) a specific multicast address.

The multicast clients need a mechanism to *join* multicast groups. **Internet Group Management Protocol (IGMP)** allows clients to send "requests" to multicast-enabled routers to join a multicast group.

IGMP only handles group membership. To actually route multicast data to a client, a multicast routing protocol is required, such as PIM or DVMRP.

Three versions of IGMP exist, **IGMPv1, IGMPv2,** and **IGMPv3**.

IGMPv1 routers send out a "query" every **60 seconds** to determine if any hosts need access to a multicast server. This query is sent out to the 224.0.0.1 address (i.e., all hosts on the subnet). Interested hosts must reply with a **Membership Report** stating what multicast group they wish to join.

Unfortunately, IGMPv1 does not allow hosts to dynamically "leave" a group. Instead, if no Membership Reports are received after 3 times the query interval, the router will flush the hosts out of its IGMP table.

IGMPv2 adds additional functionality. Queries can be sent out either as **General Queries** (224.0.0.1) or **Group-Specific Queries** (only sent to specific group members). Additionally, hosts can send a **Leave Group** message to IGMPv2 routers, to immediately be flushed out of the IGMP table. Thus, IGMPv2 allows the multicast tree to by updated more efficiently.

All versions of IGMP elect one router to be the **Designated Querier** for that subnet. The router with the **lowest IP address** becomes Designated.

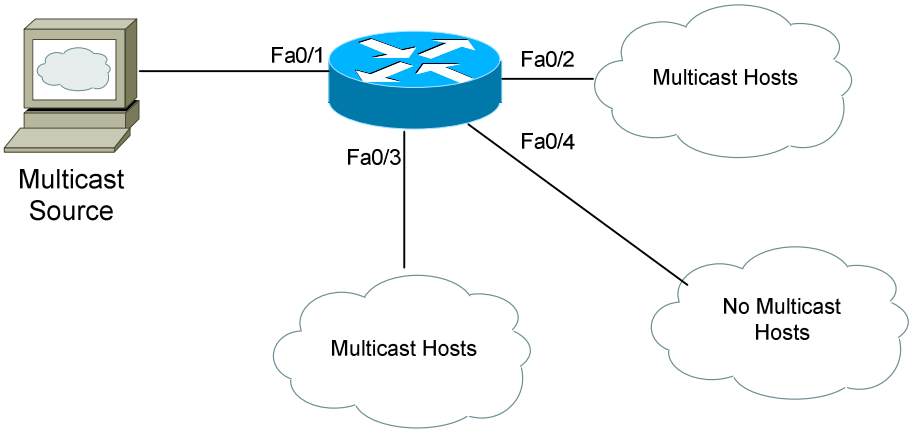
IGMPv1 is not compatible with IGMPv2. If any IGMPv1 routers exist on the network, *all* routers must operate in IGMPv1 mode.

Cisco IOS version 11.1 and later support IGMPv2 by default.

IGMPv3 enhances v2 by supporting **source-based filtering** of multicast groups. Essentially, when a host responds to an IGMP query with a Membership Report, it can specifically identify which *sources* within a multicast group to join (or even *not* join).

* * *

IGMP Example



In the above example, assume the router is using IGMPv2. Interface fa0/1 points towards the multicast source, and thus becomes the **upstream** interface.

Initially, the router will sent out Group Specific Queries out all non-upstream interfaces. Any multicast hosts will respond with a Membership Report stating what multicast group they wish to join.

Interfaces fa0/2 and fa0/3 will become **downstream** interfaces, as they contain multicast hosts. No multicast traffic will be sent out fa0/4.

If all multicast hosts leave the multicast group off of interface fa0/2, it will be removed from the multicast tree. If a multicast host is ever added off of interface fa0/4, it will become a downstream interface.

* * *

IGMP Configuration

No configuration is required to enable IGMP, except to enable IP multicast routing (*ip multicast-routing*). We can change the version of IGMP running on a particular interface (by default, it is Version 2):

Switch(config-if)# *ip igmp version 1*

To view which multicast groups the router is aware of:

Switch# *show ip igmp groups*

We can join a router interface to a specific multicast group (forcing the router to respond to ICMP requests to this multicast group):

Switch(config-if)# *ip igmp join-group 226.1.5.10*

WE can also simply force a router interface to *always* forward the traffic of a specific multicast group out an interface:

Switch(config-if)# *ip igmp static-group 226.1.5.10*

We can also restrict which multicast groups a host, off of a particular interface, can join:

Switch(config)# *access-list 10 permit 226.1.5.10*
Switch(config)# *access-list 10 permit 226.1.5.11*

Switch(config-if)# *ip igmp access-group 10*

* * *

Protocol Independent Multicast (PIM)

While IGMP concerns itself with allowing multicast hosts to join multicast groups, **Protocol Independent Multicast (PIM)** is a multicast routing protocol that is concerned about *getting* the multicast data to its destination (or, more accurately, *taking* the data away from the multicast source).

PIM is also responsible for creating the multicast tree, and "pruning" the tree so that no traffic is sent unnecessarily down a link.

PIM can operate in three separate modes:

- **PIM Dense Mode (**PIM-DM**)**
- **PIM Sparse Mode** (PIM-SM)
- **PIM Sparse-Dense Mode** (PIM-SM-DM, Cisco proprietary)

The key difference between PIM Dense and Sparse Mode is how the multicast tree is created. With **PIM Dense Mode**, all networks are flooded with the multicast traffic from the source. Afterwards, networks that don't need the multicast are pruned off of the tree. The network that contains the multicast source becomes the "root" of the multicast network.

With **PIM Sparse Mode**, no "flooding" occurs. Only networks that contain "requesting" multicast hosts are added to the multicast tree. A centralized PM router, called the **Rendezvous Point (RP),** is elected to be the "root" router of the multicast tree. PIM routers operating in Sparse Mode build their tree towards the RP, instead of towards the multicast source. The RP allows multiple multicast "sources" to utilize the same multicast tree.

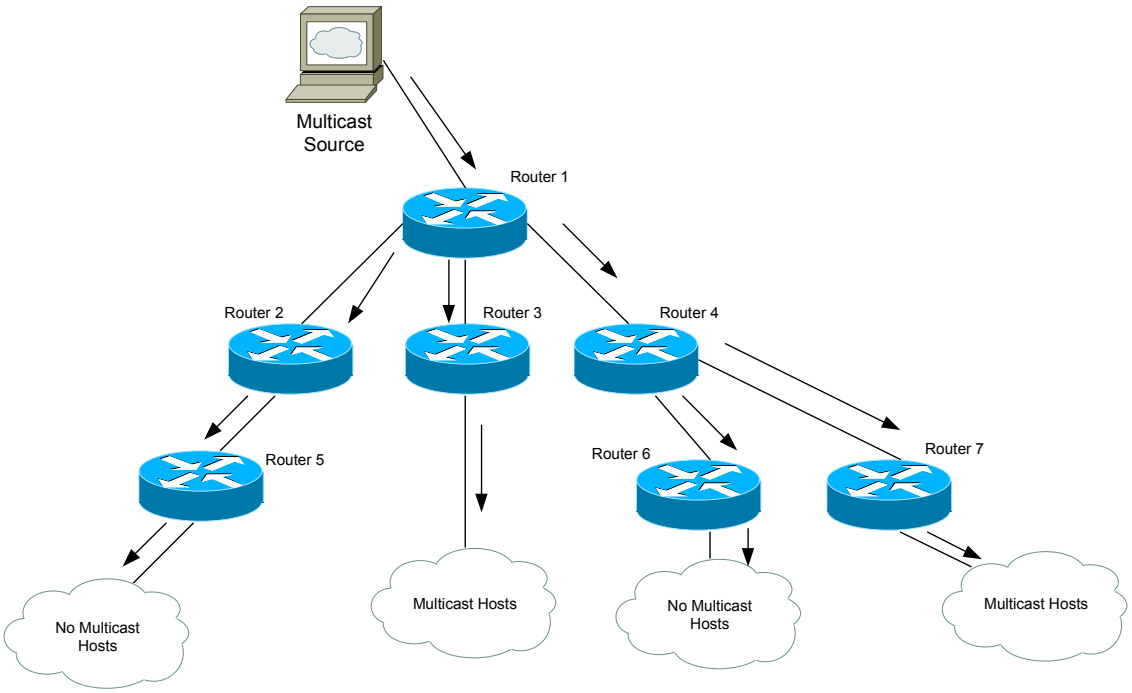
PIM Sparse-Dense Mode allows either Sparse or Dense Mode to be used, depending on the multicast group. Any group that points to an RP utilizes Sparse Mode. PIM Sparse-Dense Mode is Cisco proprietary.

Consider these key points:

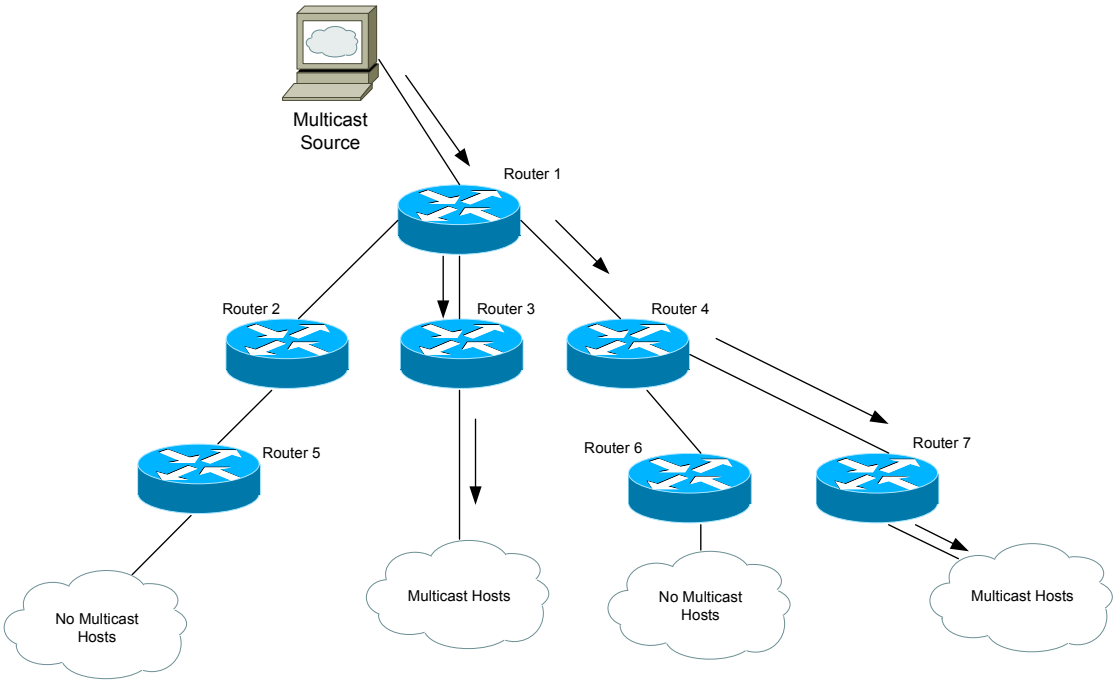
- **Dense Mode** should be used when a **large number** of multicast hosts exist across the internetwork. The "flooding" process allows for a quick creation of the multicast tree, at the expense of wasting bandwidth.
- **Sparse Mode** should be used when only a **limited number** of multicast hosts exist. Because hosts must explicitly join before that network segment is added to the multicast tree, bandwidth is utilized more efficiently.

* * *

PIM Dense Mode Example

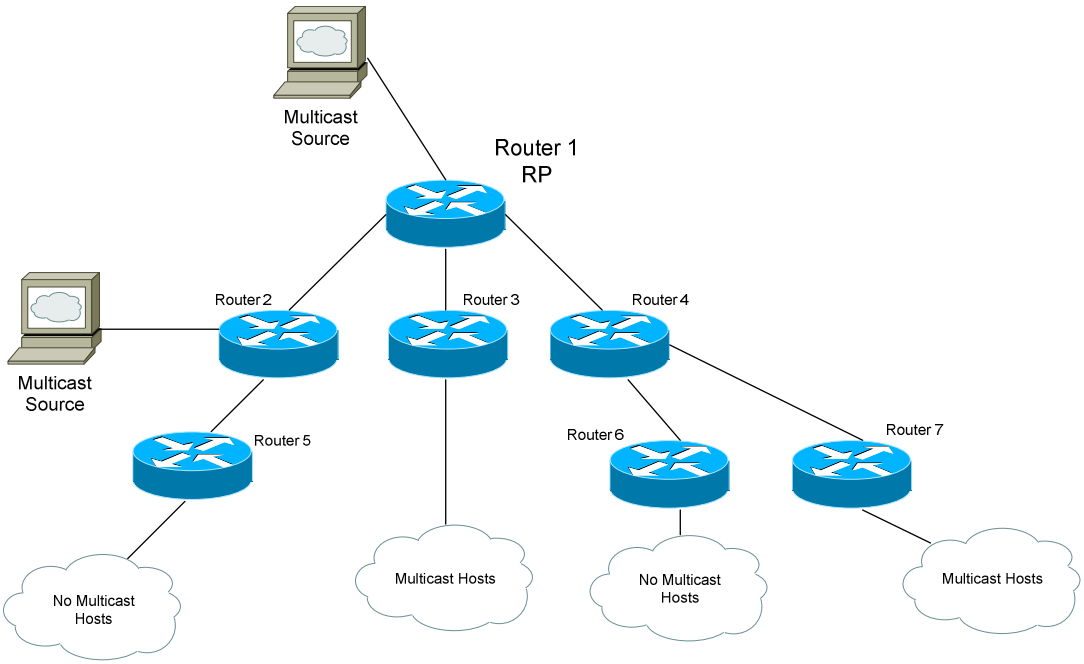


Consider the above example. When PIM routers operate in **Dense Mode**, all segments of the multicast tree are flooded initially. Eventually, "branches" that do not require the multicast traffic are pruned off:

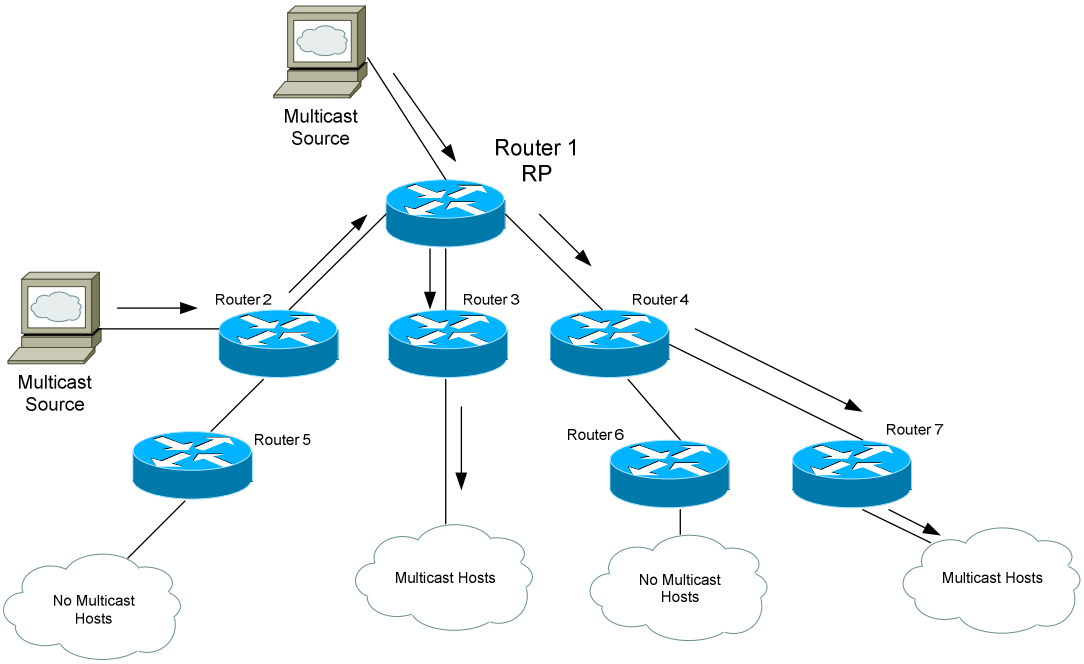


* * *

PIM Sparse Mode Example



When PIM routers operate in **Sparse Mode**, multicast traffic is *not* initially flooded throughout the entire multicast tree. Instead, a Rendezvous Point (RP) is elected or designated, and all multicast sources and clients must explicitly register with the RP. This provides a centralized method of directing the multicast traffic of multiple multicast sources:



* * *

Configuring Manual PIMv1

Two versions of PIM exist (**PIMv1** and **PIMv2**), though both are very similar. PIM must be enabled on each participating interface in the multicast tree.

To enable PIM and specify its mode on an interface:

```
Switch(config)#  interface fa0/10
Switch(config-if)#  no switchport
Switch(config-if)#  ip pim dense-mode
Switch(config-if)#  ip pim sparse-mode
Switch(config-if)#  ip pim sparse-dense-mode
```

When utilizing PIM-SM, we must configure a Rendezvous Point (RP). RP's can be identified manually, or dynamically chosen using a process called **auto-RP (**Cisco-proprietary).

To manually specify an RP on a router:

```
Switch(config)#  ip pim rp-address 192.168.1.1
```

The above command must be configured on *every* router in the multicast tree, including the RP itself.

To restrict the RP to a specific set of multicast groups:

```
Switch(config)#  access-list 10 permit 226.10.10.1
Switch(config)#  access-list 10 permit 226.10.10.2
Switch(config)#  ip pim rp-address 192.168.1.1 10
```

The first two commands create an *access-list 10* specifying the multicast groups this RP will support. The third command identifies the *RP*, and applies *access-list 10* to the RP.

* * *

Configuring Dynamic PIMv1

When using Cisco's auto-RP, one router is designated as a **Mapping Agent**. To configure a router as a mapping agent:

Switch(config)# *ip pim send-rp-discovery scope 10*

The *10* parameter in the above command is a TTL (Time to Live) setting, indicating that this router will serve as a mapping agent for up to 10 hops away.

Mapping agents listen for **candidate** RP's over multicast address 224.0.1.39 (Cisco RP Announce). To configure a router as a candidate RP:

Switch(config)# *access-list 10 permit 226.10.10.1*
Switch(config)# *access-list 10 permit 226.10.10.2*
Switch(config)# *ip pim send-rp-announce fa0/10 scope 4 group-list 10*

The first two commands create an *access-list 10* specifying the multicast groups this RP will support. The third command identifies this router as a candidate RP for the multicast groups specified in *group-list 10*. This RP's address will be based on the IP address configured on *fa0/10*. The *scope 4* parameter indicates the maximum number of hops this router will advertise itself for.

The above commands essentially create a "mapping" of specific RP's to specific multicast groups. Once a mapping agent learns of these mappings from candidate RPs, it sends the information to all PIM routers over multicast address 224.0.1.40 (Cisco RP Discovery).

Configuring Dynamic PIMv2

Configuring PIMv2 is very similar to PIMv1, except that PIMv2 is a standards-based protocol. Also, there are terminology differences. Instead of mapping agents, PIMv2 uses **Bootstrap Routers (BSR),** which performs the same function.

To configure a router as a BSR:

Switch(config)# *ip pim bsr-candidate fa0/10*

To configure candidate RP's in PIMv2:

Switch(config)# *access-list 10 permit 226.10.10.1*
Switch(config)# *access-list 10 permit 226.10.10.2*
Switch(config)# *ip pim rp-candidate fa0/10 4 group-list 10*

The first two commands create an *access-list 10* specifying the multicast groups this RP will support. The third command identifies this router as a candidate RP for the multicast groups specified in *group-list 10*. This RP's address will be based on the IP address configured on *fa0/10*. The *4* parameter indicates the maximum number of hops this router will advertise itself for.

With PIMv2, we can create **border routers** to prevent PIM advertisements (from the BSR or Candidate RPs) from passing a specific point.

To configure a router as a PIM border router:

Switch(config)# *ip pim border*

Multicasts and Layer 2 Switches

Up to this point, we've discussed how multicasts interact with routers or multilayer switches.

By default, a Layer 2 switch will forward a multicast out all ports, excluding the port it received the multicast on. To eliminate the need of "flooding" multicast traffic, two mechanisms have been developed for Layer 2 switches:

- **IGMP snooping**
- **CGMP**

IGMP snooping allows a Layer 2 switch to "learn" the multicast MAC address of multicast groups. It does this by eavesdropping on IGMP Membership Reports sent from multicast hosts to PIM routers. The Layer 2 switch then adds a multicast MAC entry in the CAM for the specific port that needs the multicast traffic.

IGMP snooping is **enabled by default** on the Catalyst 2950 and 3550. If disabled, it can be enabled with the following command:

Switch(config)# *ip igmp snooping*

If a Layer 2 switch does not support IGMP snooping, **Cisco Group Membership Protocol (CGMP)** can be used. Three guesses as to whether this is Cisco-proprietary or not.

Instead of the Layer 2 switch "snooping" the IGMP Membership Reports, CGMP allows the PIM router to actually **inform** the Layer 2 switch of the multicast MAC address, and the MAC of the host joining the group. The Layer 2 switch can then add this information to the CAM.

CGMP must be configured on the **PIM router** (or **multilayer switch**). It is **disabled** by default on all PIM routers. To enable CGMP:

Switch(config-if)# *ip cgmp*

No configuration needs to occur on the Layer 2 switch.

* * *

Troubleshooting Multicasting

To view IGMP groups and current members:

Switch# *show ip igmp groups*

To view the IGMP snooping status:

Switch# *show ip igmp snooping*

To view PIM "neighbors":

Switch# *show ip pim neighbor*

To view PIM RPs:

Switch# *show ip pim rp*

To view PIM RP-to-Group mappings:

Switch# *show ip pim rp mapping*

To view the status of PIMv1 Auto-RP:

Switch# *show ip pim autorp*

To view PIMv2 BSRs:

Switch# *show ip pim bsr-router*

We can also debug multicasting protocols:

Switch# *debug ip igmp*
Switch# *debug ip pim*

* * *

Viewing the Multicast Table

Just like unicast routing protocols (such as OSPF, RIP), multicast routing protocols build a routing table.

Again, these tables contain several elements:

- The multicast **source**, and its associated multicast address (labeled as "**S,G**", or "**Source,Group")**
- **Upstream** interfaces that point *towards* the source
- **Downstream** interfaces that point *away* from the source towards multicast hosts.

To view the multicast routing table:

Switch# *show ip mroute*

If using PIM in **Dense Mode**, the output would be similar to the following:

```
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(10.1.1.1/24, 239.5.222.1), uptime 1:11:11, expires 0:04:29, flags: C
  Incoming interface: Serial0, RPF neighbor 10.5.11.1
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 2:52:11/0:01:12
```

Remember that a multicast source with its associated multicast address is labeled as (S,G). Thus, in the above example, 10.1.1.1/24 is the multicast source, while 239.5.222.1 is the multicast address/group that the source belongs to.

The Incoming interface indicates the **upstream** interface. The RPF neighbor is the next hop router "upstream" towards the source. The outgoing interface(s) indicate **downstream** interfaces.

Notice that the **S – Sparse** flag is not set. That's because PIM is running in Dense Mode.

Viewing the Multicast Table (continued)

Remember, to view the multicast routing table:

Switch# *show ip mroute*

If using PIM in **Sparse Mode**, the output would be similar to the following:

```
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.59.222.10), uptime 2:11:05, RP is 10.1.1.10, flags: SC
  Incoming interface: Serial0, RPF neighbor 10.3.35.1,
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 4:41:22/0:05:21
```

Notice that the (S,G) pairing is labeled as (*, 224.59.222.10). In Sparse Mode, we can have *multiple* sources share the same multicast tree.

The Rendezvous Point (RP) is 10.1.1.10. The flags are set to **SC**, indicating this router is running in Sparse Mode.

Just like with Dense Mode, the Incoming interface indicates the **upstream** interface, and the outgoing interface(s) indicate **downstream** interfaces.

However, the RPF neighbor is the next hop router "upstream" towards the **RP** now, and not the source.