IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A PRACTICAL STUDY OF MULTICAST ROUTING: IGMP VERSIONS AND PIM MODES WITH LAB IMPLEMENTATION

Vivek K R
Senior Technology Operations Engineer
Lowe's India Private Limited, Bengaluru, India

Abstract: Multicast is used in networking to efficiently send data from one sender to multiple receivers, without burdening the network with multiple copies of the same data. It helps in efficient usage of bandwidth instead of sending individual copies of the same data to each receiver (as with unicast), multicast sends a single copy of the data to all receivers that are interested. This saves bandwidth, especially when there are many receivers. Sending a video stream to multiple devices can be done using broadcast, but a broadcast packet sends traffic to all ports and will not be forwarded across routers. To overcome this limitation, we use multicast. This paper provides a brief overview of the components of multicast and the protocols that help multicast to work

Introduction

In this paper, the importance of IGMP, the differences between IGMP Version 1, Version 2, and Version 3, as well as multicast routing with PIM Dense Mode and PIM Sparse Mode and their differences, will be discussed. Everything will be explained with the help of labs, and the working of multicast flow will be explored. The components needed for multicast to work are Class D IP addresses (224.0.0.0 to 239.255.255.255), applications like VLC player, IGMP (the protocol that helps the client inform the router to receive multicast traffic), IGMP Snooping (which allows the switch to figure out where the packet needs to be sent). The switch forwards the packet based on the MAC address. The switch learns MAC addresses using source learning; however, the multicast MAC address is in the destination, so the switch will never learn it. To fix this issue, IGMP snooping is used. The next component is multicast routing, specifically PIM, which is the mechanism used to route multicast packets to a destination (a multicast group address).

IGMP (Internet group management protocol)

IGMP (Internet Group Management Protocol) is a communication protocol used by hosts and routers to manage the membership of IP multicast groups. It allows devices on a network to report their interest in receiving multicast traffic for a specific multicast group address. IGMP (Internet Group Management Protocol) uses two main types of messages to manage multicast group membership: **Query messages** and **Report messages.**

The **IGMP Query** message is sent by routers to ask hosts if they want to join a specific multicast group. It's a way for routers to check which devices are interested in receiving multicast traffic for a given multicast group.

The **IGMP Report** message is sent by hosts (or devices) to notify routers that they are interested in receiving traffic for a specific multicast group. When a host receives an IGMP Query and wants to join the group, it sends an IGMP Report message back to the router.

To demonstrate IGMP Versions 1, 2, and 3, the topology shown in figure 1 is used. In this setup, a multicast-enabled router, referred to as the multicast source, is connected to a Layer 2 switch, which in turn connects to a multicast host that requests multicast traffic

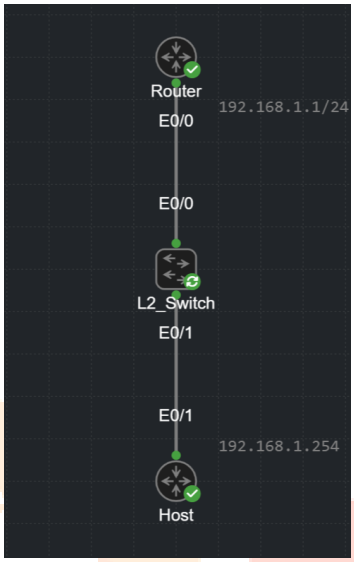

Figure 1

In this setup, IP multicast routing is enabled on the multicast router, and Interface Ethernet 0/0 is configured to use IGMP Version 1. By default, IGMP Version 2 is used. PIM Sparse Mode is also enabled and will be explained in detail in the following sections

Router(config)#*ip multicast-routing*
Router(config)#*int eth0/0*
Router(config-if)#*ip igmp version 1*
Router(config-if)#*ip pim sparse-mode*

Let's verify the configuration

Router#*show ip igmp int eth0/0*
Ethernet0/0 is up, line protocol is up
Internet address is 192.168.1.1/24
IGMP is enabled on interface
Current IGMP host version is 1
Current IGMP router version is 1
IGMP query interval is 60 seconds
IGMP configured query interval is 60 seconds
Inbound IGMP access group is not set
IGMP activity: 1 joins, 0 leaves
Multicast routing is enabled on interface
Multicast TTL threshold is 0
Multicast designated router (DR) is 192.168.1.1 (this system)
IGMP querying router is 192.168.1.1 (this system)
Multicast groups joined by this system (number of users):
224.0.1.40(1)

In this setup, a router was configured to function as the multicast host (named Host). IGMP Version 1 was enabled on Interface Ethernet0/1, and the *debug ip igmp* command was executed on both the host and the upstream router. This enabled real-time monitoring of IGMP query and report messages, allowing for detailed analysis of multicast group membership behavior.

```
Host(config)#int eth0/1
Host(config-if)#ip igmp version 1
Host(config-if)#ip igmp join-group 239.1.1.1

Router#debug ip igmp
IGMP debugging is on
*Mar 26 16:23:34.314: IGMP(0)[default]: Send v1 general Query on Ethernet0/0
*Mar 26 16:23:34.314: IGMP(0)[default]: Set report delay time to 0.5 seconds for 224.0.1.40 on
Ethernet0/0
*Mar 26 16:23:34.909: IGMP(0)[default]: Send v1 Report for 224.0.1.40 on Ethernet0/0
*Mar 26 16:23:34.909: IGMP(0)[default]: Received v1 Report on Ethernet0/0 from 192.168.1.1 for
224.0.1.40
```

The report and query messages can be observed. Now, suppose the host is no longer interested in receiving the multicast traffic

```
Host(config-if)#no ip igmp join-group 239.1.1.1
*Mar 26 16:33:41.991: IGMP(0)[default]: IGMP delete group 239.1.1.1 on Ethernet0/1
*Mar 26 16:33:41.993: IGMP(0)[default]: Deregistering IGMP with ipmulticast GIR
```

The debug logs indicate that the IGMP group has been deleted. However, the router will continue forwarding multicast traffic until the corresponding timer expires

```
Router#show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address   Interface          Uptime    Expires   Last Reporter   Group Accounted
239.1.1.1       Ethernet0/0        00:11:24  00:02:18  192.168.1.2
```

In IGMP Version 1, if the host stops listening to multicast traffic, it will not reply to the router. This issue is fixed in Version 2 with the **Leave Group** message

```
Router(config-if)#ip igmp version 2
```

After enabling IGMP Version 2, multicast traffic is visible on the router as expected.

```
Router#show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address   Interface          Uptime    Expires   Last Reporter   Group Accounted
239.1.1.1       Ethernet0/0        00:03:07  00:02:09  192.168.1.2
```

Next, the leave group message feature in IGMP Version 2 will be examined.

```
Host(config-if)#no ip igmp join-group 239.1.1.1
Host(config-if)#
*Mar 26 16:48:18.659: IGMP(0)[default]: IGMP delete group 239.1.1.1 on Ethernet0/1
*Mar 26 16:48:18.660: IGMP(0)[default]: Send Leave for 239.1.1.1 on Ethernet0/1
```

After the host is not interested in the traffic the multicast traffic is immediately removed.

```
Router#show ip igmp groups 239.1.1.1
IGMP Connected Group Membership
Group Address    Interface            Uptime    Expires   Last Reporter   Group Accounted
```

The main advantage of IGMP Version 3 (IGMPv3) is support for Source-Specific Multicast (SSM). This allows hosts to join a multicast group and specify the particular source(s) from which they want to receive traffic

```
Router(config)#int eth0/0
Host(config-if)#ip igmp version 3
Host(config-if)#ip igmp join-group 239.1.1.1 source 1.1.1.1
```

The debug output indicates that the router is creating a source-specific entry for 1.1.1.1. As a result, the host will now receive multicast traffic exclusively from the source 1.1.1.1.

```
*Mar 26 16:58:12.278: IGMP(0)[default]: Received v3 Report for 1 group on Ethernet0/0 from 0.0.0.0
*Mar 26 16:58:12.279: IGMP(0)[default]: Received Group record for group 239.1.1.1, mode 5 from
0.0.0.0 for 1 sources
*Mar 26 16:58:12.279: IGMP(0)[default]: Create source 1.1.1.1
*Mar 26 16:58:12.279: IGMP(0)[default]: Updating expiration time on (1.1.1.1,239.1.1.1)   to 180 secs
*Mar 26 16:58:12.279: IGMP(0)[default]: Setting source flags 4 on (1.1.1.1,239.1.1.1)
*Mar 26 16:58:12.279: IGMP(0)[default]: MRT Add/Update Ethernet0/0 for (*,239.1.1.1) by 0
```

IGMP Snooping

The switch will forward packets based on the MAC address. The switch learns the MAC address from the source field, but the multicast MAC address will be on the destination side, so the switch will never learn the multicast MAC address. IGMP snooping helps resolve this issue.

Snooping refers to the switch "listening" in on the IGMP messages exchanged between devices and the router. When a device sends an IGMP **Join message** to request multicast traffic, the switch **"snoops"** on that message. The switch learns which ports have devices that want to receive traffic for specific multicast groups.

If the switch hears an IGMP Leave message, indicating a device no longer wants the multicast stream, it removes the device from the list. IGMP is not a standard RFC; Cisco's way is IGMP snooping
In the lab, the same topology as shown in figure 1 is used. By default, snooping is enabled on the switch.

debug ip igmp snooping router is enabled on the switch (L2_Switch). Once the query packet is sent from the router, the L2_Switch recognizes that interface Ethernet 0/0 is connected to the router, since only the router sends IGMP query packets.

```
L2_Switch#
*Apr 22 22:58:37.115: IGMPSN: router: Received IGMP pak on Vlan 1, port Et0/0
*Apr 22 22:58:37.115: IGMPSN: router: port Et0/0 is a router port on Vlan 1
*Apr 22 22:58:37.115: IGMPSN: router: Learning port: Et0/0 as rport on Vlan 1
L2_Switch#
*Apr 22 22:58:44.210: IGMPSN: router: port Et0/0 is a router port on Vlan 1

L2_Switch#show ip igmp snooping querier
Vlan      IP Address           IGMP Version  Port
-----------------------------------------------------------
1         192.168.1.1          v2            Et0/0
```

At this stage, the command ip join-group 239.1.1.1 is configured on Host interface Ethernet 0/1, indicating its interest in receiving multicast traffic for group 239.1.1.1. Following this, the exchange of IGMP query and report packets is initiated. The switch connected to Host (Ethernet 0/1) detects the multicast group 239.1.1.1 on Interface Ethernet 0/1 of the layer 2 switch (L2_Switch). This signifies that the host connected to port Ethernet 0/1 has requested multicast traffic for the group 239.1.1.1. In this way, the switch understands that traffic destined for the group 239.1.1.1 should be forwarded to interface Ethernet 0/1 of the L2_Switch, where the host is connected.

***L2_Switch**#show ip igmp snooping groups*
Flags: I -- IGMP snooping, S -- Static, P -- PIM snooping, A -- ASM mode

Vlan	*Group/source*	*Type*	*Version*	*Port List*
1	*224.0.1.40*	*I*	*v2*	*Et0/1*
1	***239.1.1.1***	***I***	***v2***	***Et0/1***

Multicast Routing

In unicast routing, a packet is forwarded based on the routing entry found in the router's routing table. The routing table contains information on how to reach various networks or hosts. It has entries that map destination IP ranges (subnets) to the appropriate next-hop router or outgoing interface. However, in multicast routing, the destination field will be the multicast group address, and this multicast group address will not be in the unicast routing table. Therefore, a different routing protocol is needed to route the multicast address.

PIM is the most widely used protocol for multicast routing. It's called "Protocol Independent" because it doesn't rely on any specific unicast routing protocol (e.g., RIP, OSPF) but instead uses the unicast routing table as a reference for the best path. PIM can operate in two primary modes: PIM-Dense Mode and PIM-Sparse Mode.

PIM-Dense Mode

PIM Dense Mode, multicast traffic is flooded throughout the network, and routers only prune back the branches of the multicast distribution tree where there are no receivers.

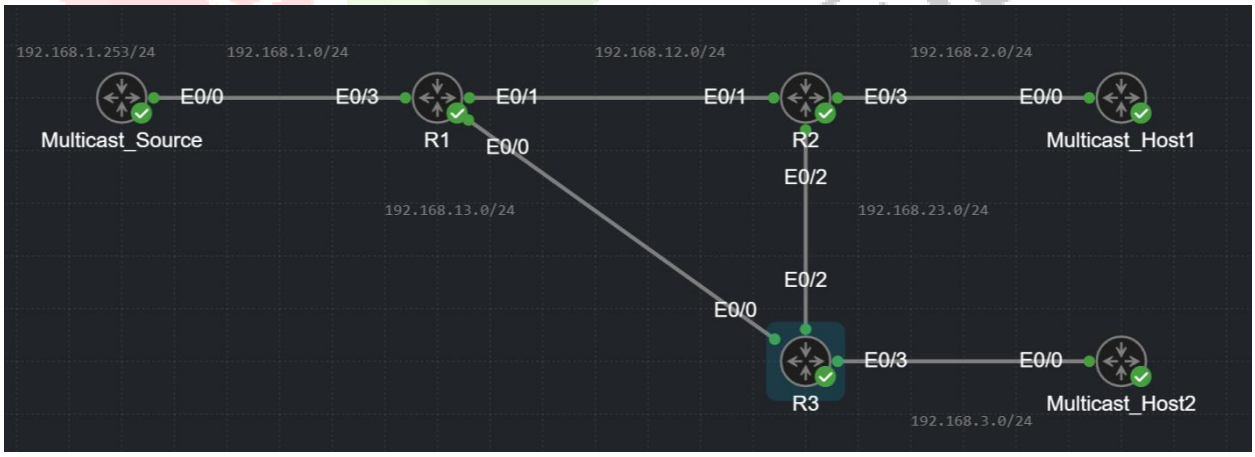

Figure 2

So, in this example, the Multicast_Source is sending multicast packets towards the R1. As soon as R1 receives the multicast packet, it will create an entry in its routing table, storing the source address (in this case, 192.168.1.253) and the multicast group 239.1.1.1 in its multicast routing table. The packet will then be flooded from R1 to all other interfaces (Ethernet0/1 and Ethernet0/0), except for the interface where it was received (Ethernet0/3). And this process continues on R2 and R3. This creates a high chance of forming a loop. To avoid a loop, there is an interesting method called RPF (Reverse Path Forwarding). In the example in figure 2, R3 will

receive multicast packets from both R2 and R1. R3 needs to decide which packet to accept and which to drop. R3 will examine the multicast packet, check the source address (192.168.1.253), and then look up the source address (192.168.1.253) in the unicast routing table to determine the outgoing interface that leads to the multicast source. If the packet arrives on the correct interface (i.e., the one that matches the source address in the routing table), it is forwarded as usual. If it doesn't match, the packet is discarded. R3 will also send a prune packet to the router from which the multicast packet was discarded. The prune packet indicates that a certain branch or path in the multicast distribution tree no longer needs to receive multicast traffic

Proceeding with the lab setup. For multicast to function properly, unicast routing must be enabled on all routers. In this scenario, OSPF is configured between all routers. The multicast source and multicast hosts are represented by routers within the lab. On R1, R2, and R3, the unicast routing is configured.

```
R1#show ip route ospf | b 192
O    192.168.2.0/24 [110/20] via 192.168.12.2, 00:24:29, Ethernet0/1
O    192.168.3.0/24 [110/20] via 192.168.13.2, 00:24:07, Ethernet0/0
O    192.168.23.0/24 [110/20] via 192.168.13.2, 00:24:07, Ethernet0/0
[110/20] via 192.168.12.2, 00:23:33, Ethernet0/1

R2#show ip route ospf | b 192
O    192.168.1.0/24 [110/20] via 192.168.12.1, 00:25:36, Ethernet0/1
O    192.168.3.0/24 [110/20] via 192.168.23.2, 00:24:40, Ethernet0/2
O    192.168.13.0/24 [110/20] via 192.168.23.2, 00:24:40, Ethernet0/2
          [110/20] via 192.168.12.1, 00:25:36, Ethernet0/1

R3#show ip route ospf | b 192
O    192.168.1.0/24 [110/20] via 192.168.13.1, 00:25:54, Ethernet0/0
O    192.168.2.0/24 [110/20] via 192.168.23.1, 00:25:20, Ethernet0/2
O    192.168.12.0/24 [110/20] via 192.168.23.1, 00:25:20, Ethernet0/2
          [110/20] via 192.168.13.1, 00:25:54, Ethernet0/0
```

The next step is to enable IP multicast routing on all routers and activate PIM Dense-Mode on the interfaces

```
R1(config)#ip multicast-routing
R1(config)#int ethernet0/1
R1(config-if)#ip pim dense-mode
```

Once PIM Dense Mode is enabled, PIM neighborship formation can be observed.

```
R1#show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface            Uptime/Expires    Ver   DR
Address                                                        Prio/Mode
192.168.13.2      Ethernet0/0          00:35:39/00:01:25 v2    1 / DR S P G
192.168.12.2      Ethernet0/1          00:36:05/00:01:34 v2    1 / DR S P G
```

Now, the multicast routing table of R1 will be examined.

```
R1#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
```

T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
** - determined by Assert, # - iif-starg configured on rpf intf,*
e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
t - LISP transit group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:07:53/00:02:07, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/0, Forward/Dense, 00:03:08/stopped, flags:
Ethernet0/1, Forward/Dense, 00:07:53/stopped, flags:

At this moment, there is no multicast traffic, but we are seeing 224.0.1.40 in the multicast routing table. This might be confusing—it corresponds to the Auto-Rendezvous Point (Auto-RP), which is related to PIM Sparse Mode. We will discuss that later. PIM Dense Mode does not use any type of Rendezvous Point.

Now, let's log in to Multicast_Host1 and enable IGMP join for the group 239.1.1.1.

Multicast_Host1(config)#*interface ethernet0/0*
Multicast_Host1(config-if)#*ip igmp join-group 239.1.1.1*

Login to R2 and check the multicast route. An entry for the group 239.1.1.1 is visible, as Multicast_Host1 is attempting to join the group 239.1.1.1. R2 adds this group to the multicast routing table. However, traffic from the source has not been received yet, therefore, a '*' appears in the multicast routing table. As a result, the multicast packet cannot be forwarded.

R2*#show ip mroute*
IP Multicast Routing Table
(*, 239.1.1.1), 01:01:49/00:02:07, RP 0.0.0.0, flags: DC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/1, Forward/Dense, 01:01:30/stopped, flags:
Ethernet0/2, Forward/Dense, 01:01:33/stopped, flags:
Ethernet0/3, Forward/Dense, 01:01:49/stopped, flags:
(*, 224.0.1.40), 01:01:59/00:02:15, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/2, Forward/Dense, 01:01:33/stopped, flags:
Ethernet0/1, Forward/Dense, 01:01:59/stopped, flags:

Generate traffic from the multicast source by pinging the multicast group IP 239.1.1.1 from Multicast_Source.

Multicast_Source*#ping 239.1.1.1 r 500*
Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:
Reply to request 1 from 192.168.2.253, 1 ms
Reply to request 2 from 192.168.2.253, 1 ms

Now in the multicast route of R1 we will be able see the new entry i.e. the source IP 192.168.1.253. (192.168.1.253, 239.1.1.1), entry in the multicast routing table is the (S,G) entry. The T flag indicates the use of the shortest path tree. The packet is being forwarded to Ethernet 0/1. Since R3 does not have any active multicast traffic, Ethernet 0/0 on R1 will be pruned

R1*#show ip mroute 239.1.1.1*
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
** - determined by Assert, # - iif-starg configured on rpf intf,*
e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
t - LISP transit group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 00:08:07/stopped, RP 0.0.0.0, flags: D
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/1, Forward/Dense, 00:08:07/stopped, flags:
Ethernet0/0, Forward/Dense, 00:08:07/stopped, flags:

(192.168.1.253, 239.1.1.1), 00:08:07/00:02:46, flags: T
Incoming interface: Ethernet0/3, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/0, Prune/Dense, 00:02:10/00:00:55, flags:
Ethernet0/1, Forward/Dense, 00:08:07/stopped, flags*:*

Assume there is a change in the network, with Multicast_Host2 requesting a multicast packet. In this case, Ethernet0/2 of R2 or R1's Ethernet 0/0 needs to change from prune to forward/dense. This unpruning is achieved using the graft message.

Enable debugging

```
R3#debug ip pim
PIM debugging is on
```

Log in to Multicast_Host2 and enable the IGMP join for the group 239.1.1.1.

The GRAFT message is observed

```
Multicast_Host2(config)#int ethernet0/0
Multicast_Host2(config-if)#ip igmp join-group 239.1.1.1
*Mar 25 16:06:59.813: PIM(0)[default]: Building Graft message for 239.1.1.1, Ethernet0/3: no entries
*Mar 25 16:06:59.813: PIM(0)[default]: Building Graft message for 239.1.1.1, Ethernet0/0:
 192.168.1.253/32 count 1
*Mar 25 16:06:59.813: PIM(0)[default]: Send v2 Graft to 192.168.13.1 (Ethernet0/0)
*Mar 25 16:06:59.813: PIM(0)[default]: Building Graft message for 239.1.1.1, Ethernet0/2: no
entries
*Mar 25 16:06:59.814: PIM(0)[default]: Received v2 Graft-Ack on Ethernet0/0 from 192.168.13.1
*Mar 25 16:06:59.814:      Group 239.1.1.1:
```

Ethernet 0/0 is now in a forward state, meaning it is forwarding multicast traffic.

```
R1#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf,
       e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                          t - LISP transit group
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 00:34:54/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Dense, 00:34:54/stopped, flags:
    Ethernet0/0, Forward/Dense, 00:34:54/stopped, flags:

(192.168.1.253, 239.1.1.1), 00:34:54/00:00:43, flags: T
  Incoming interface: Ethernet0/3, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Dense, 00:10:53/stopped, flags:
    Ethernet0/1, Forward/Dense, 00:34:54/stopped, flags:
```

PIM-Dense Mode (PIM-DM) is a multicast routing protocol best suited for environments with a high density of

multicast receivers. It operates using a flood-and-prune mechanism, initially flooding multicast traffic throughout the network and then pruning paths without receivers. While simple to deploy in smaller or tightly knit networks, PIM-DM can lead to inefficient bandwidth usage in large or sparse networks. For scalable and efficient multicast routing in modern networks, PIM-Sparse Mode or other optimized solutions are generally preferred.

PIM-Sparse Mode

PIM-Sparse Mode (PIM-SM) is a scalable and efficient multicast routing protocol designed for networks where multicast receivers are sparsely distributed. Unlike Dense Mode, PIM-SM uses a pull-based approach, sending multicast traffic only to routers that explicitly request it.

But the problem here is how does the router knows to find the source of the multicast traffic, to solve this sparse mode brings the concept of Rendezvous point. Every router that receives the multicast traffic will sent to the Rendezvous point, each router that wants the multicast traffic will also reach the Rendezvous point. We solved one problem. Next, we need to address how the routers know who the Rendezvous Point (RP) is. There are two approaches: manual and dynamic. In our lab, we are using the manual approach.

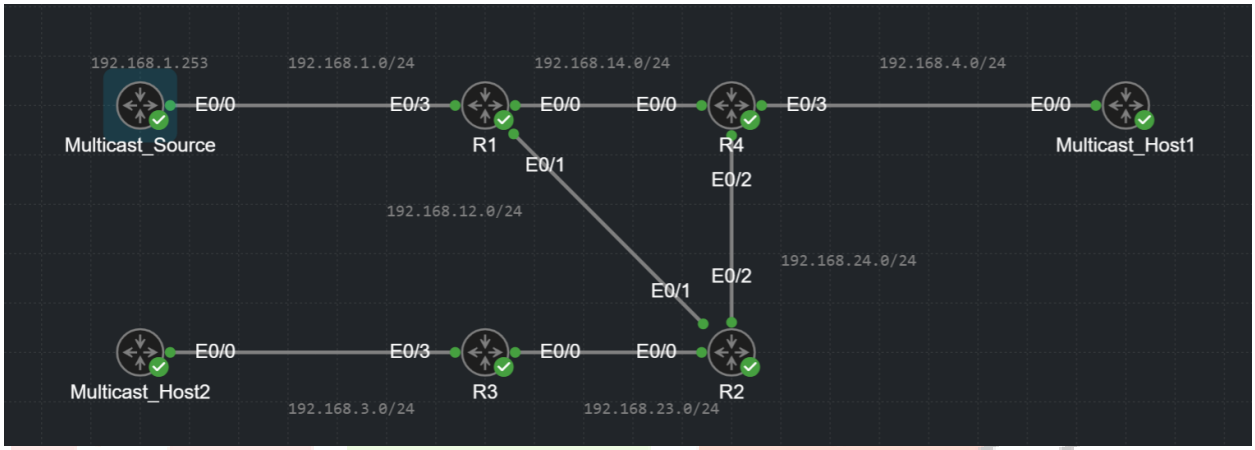

Figure 3

The network diagram in figure 3 is used to explain PIM-Sparse mode. OSPF serves as the underlying routing protocol, and multicast along with PIM-Sparse Mode is enabled at the interface level.

R1(config)#*ip multicast-routing*
R1(config)#*int eth0/1*
R1(config-if)#*ip pim sparse-mode*
R1(config-if)#*int eth0/0*
R1(config-if)#*ip pim sparse-mode*
R1(config-if)#*int eth0/3*
R1(config-if)#*ip pim sparse-mode*

Multicast routing and PIM-Sparse mode are enabled on R1, R2, R3, and R4.

Next, the Auto-RP feature is disabled, and the Rendezvous Point (RP) is manually enabled. In this case, R2 is configured as the RP by creating a loopback interface and assigning 2.2.2.2/32 as the RP address.

R2(config)#*no ip pim autorp*
R2(config)#*int loopback 0*
R2(config-if)#*ip add 2.2.2.2 255.255.255.255*

The next step is to manually configure the Rendezvous Point address on R1, R2, R3, and R4.

R2(config)#*ip pim rp-address 2.2.2.2*
**Apr 17 23:46:02.588: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up*
**Apr 17 23:46:02.589: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to up*

Two tunnels are formed on the RP router—one for encapsulation and another for decapsulation. On non-RP routers, only one tunnel is formed. The purpose of the tunnel is to encapsulate the first multicast packet in the PIM Register message and forward it to the RP.

R1*#show ip pim tunnel*
Tunnel0
Type : PIM Encap
RP : 2.2.2.2
Source : 192.168.12.1
State : UP
Last event : Created (00:11:36)

R2*#show ip pim tunnel*
Tunnel0
Type : PIM Encap
*RP : 2.2.2.2**
Source : 192.168.24.2
State : UP
Last event : Created (00:08:49)

Tunnel1*
Type : PIM Decap
*RP : 2.2.2.2**
Source : -
State : UP
Last event : Created (00:08:49)

Debugging is enabled on the routers.
In this case, traffic for the multicast group 239.4.4.4 is generated, but no interested receivers are present

Multicast_Source*#ping 239.4.4.4 repeat 500*
Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 239.4.4.4, timeout is 2 seconds:

Here's what happens: as soon as R1 receives the multicast traffic, it encapsulates the first packet into a PIM Register message and forwards it to the RP

R1*#*
**Apr 18 00:01:00.090: PIM(0)[default]: Adding register encap tunnel (Tunnel0) as forwarding interface of (192.168.1.253, 239.4.4.4).*
****Apr 18 01:34:47.356: PIM(0)[default]: Send v2 join/prune to 192.168.12.2 (Ethernet0/1)***

When the Rendezvous Point (RP) receives a PIM Register message, it has two options. If there are no interested receivers for the multicast group, the RP responds with a PIM Register-Stop message. However, if receivers are present, the RP accepts the PIM Register message. In this case, the logs indicate that a PIM Register-Stop message was sent because no hosts expressed interest in the multicast traffic

R1#
**Apr 18 00:05:00.053: PIM(0)[default]: Send v2 Data-header Register to 2.2.2.2 for 192.168.1.253, group 239.4.4.4*
****Apr 18 00:05:00.053: PIM(0)[default]: Received v2 Register-Stop on Ethernet0/1 from 2.2.2.2***

After receiving the PIM Register-Stop message, R1 remains silent for 60 seconds. Once the timer expires, it sends another PIM Register message this time as a Null Register (without data). This process repeats every 60 seconds.

Next, assume that Multicast_Host1 expresses interest in receiving traffic for multicast group 239.4.4.4. As a result, Multicast_Host1 sends an IGMP join message (i.e., a membership report) to router R4. Upon receiving this, R4 generates a PIM Join packet and forwards it toward the Rendezvous Point (RP), which in this scenario is R2. When R2 receives the PIM Join, it updates its multicast routing table. The next time R2 receives a PIM Register message from R1 via interface Ethernet0/1, it responds by sending a PIM Join message toward R1 on the same interface. This establishes the Root Path Tree (RPT), which is highlighted in figure 4.

Figure 4

Verify this in the lab by generating multicast traffic towards the multicast group 239.4.4.4

Multicast_Source*#ping 239.4.4.4 r 40*
Type escape sequence to abort.
Sending 40, 100-byte ICMP Echos to 239.4.4.4, timeout is 2 seconds:
Reply to request 2 from 192.168.4.253, 1 ms
Reply to request 3 from 192.168.4.253, 1 ms
Reply to request 4 from 192.168.4.253, 1 ms

Now the Multicast_Host1 is interested in the traffic 239.4.4.4

Multicast_Host1(config)#*int eth0/0*
Multicast_Host1(config-if)#*ip igmp join-group 239.4.4.4*

Login to R4 and checking its multicast routing table reveals that the incoming interface for multicast traffic is Ethernet0/2, which is connected to the RP

R4*#show ip mroute 239.4.4.4*
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,

```
        Z - Multicast Tunnel, z - MDT-data group sender,
        Y - Joined MDT-data group, y - Sending to MDT-data group,
        G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
        N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
        Q - Received BGP S-A Route, q - Sent BGP S-A Route,
        V - RD & Vector, v - Vector, p - PIM Joins on route,
        x - VxLAN group, c - PFP-SA cache created entry,
        * - determined by Assert, # - iif-starg configured on rpf intf,
        e - encap-helper tunnel flag, l - LISP decap ref count contributor
    Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                              t - LISP transit group
     Timers: Uptime/Expires
     Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.4.4.4), 00:06:12/00:02:34, RP 2.2.2.2, flags: SC
  Incoming interface: Ethernet0/2, RPF nbr 192.168.24.2
  Outgoing interface list:
    Ethernet0/3, Forward/Sparse, 00:06:12/00:02:34, flags:
```

Similarly, checking the multicast routing table on the RP (i.e., R2) shows that the incoming interface for multicast traffic is Ethernet0/1, and the outgoing interface is Ethernet0/2.

```
R2#show ip mroute 239.4.4.4
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf,
       e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                          t - LISP transit group
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.4.4.4), 00:08:02/00:02:40, RP 2.2.2.2, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse, 00:07:43/00:02:40, flags:

(192.168.1.253, 239.4.4.4), 00:08:02/00:03:28, flags: T
  Incoming interface: Ethernet0/1, RPF nbr 192.168.12.1
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse, 00:07:43/00:02:40, flags:
```

From this, it can be concluded that the traffic is flowing through the Root Path Tree, as shown in figure 4. Once the receiver starts receiving multicast traffic, and if the receiver's router R4 determines that a shorter (more efficient) path exists directly to the source, it may trigger a switch to the **Shortest Path Tree (SPT)**. This transition helps optimize multicast forwarding by reducing latency and bandwidth usage. In R4's routing table, the path to reach the source 192.168.1.253 is via Ethernet0/0.

```
R4#show ip route 192.168.1.253
Routing entry for 192.168.1.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Last update from 192.168.14.1 on Ethernet0/0, 01:53:31 ago
  Routing Descriptor Blocks:
  * 192.168.14.1, from 192.168.14.1, 01:53:31 ago, via Ethernet0/0
      Route metric is 20, traffic share count is 1
```

To switch from the Root Path Tree to the Shortest Path Tree, R4 sends a PIM Join message towards Ethernet0/0 and a Prune message towards Ethernet0/2

Now, checking the multicast routing table on R4 shows that the incoming interface for multicast traffic is Ethernet0/0.

```
R4#show ip mroute 239.4.4.4
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf,
       e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                          t - LISP transit group
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode


(*, 239.4.4.4), 00:59:53/stopped, RP 2.2.2.2, flags: SJC
  Incoming interface: Ethernet0/2, RPF nbr 192.168.24.2
  Outgoing interface list:
 Ethernet0/3, Forward/Sparse, 00:59:53/00:02:50, flags:

(192.168.1.253, 239.4.4.4), 00:00:42/00:02:17, flags: JT
  Incoming interface: Ethernet0/0, RPF nbr 192.168.14.1
  Outgoing interface list:
    Ethernet0/3, Forward/Sparse, 00:00:42/00:02:50, flags:
```

In R2's multicast routing table, the outgoing interface is set to Null, meaning no multicast traffic is being sent through that interface.

***R2**#show ip mroute 239.4.4.4*
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
** - determined by Assert, # - iif-starg configured on rpf intf,*
e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
t - LISP transit group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(, 239.4.4.4), 00:59:57/stopped, RP 2.2.2.2, flags: S*
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Ethernet0/2, Forward/Sparse, 00:59:37/00:02:54, flags:

(192.168.1.253, 239.4.4.4), 00:00:26/00:02:40, flags: PT
Incoming interface: Ethernet0/1, RPF nbr 192.168.12.1
Outgoing interface list: Null

Now, the traffic is flowing through the Shortest Path Tree, as shown in figure 5

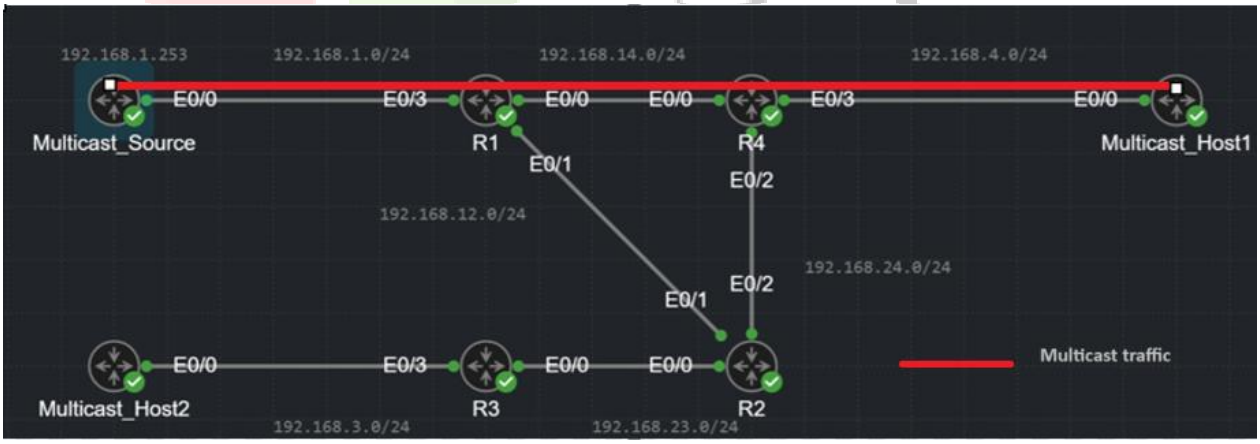

Figure 5

Conclusion

This paper provided a comprehensive overview of multicast routing protocols and mechanisms, including IGMP Versions 1, 2, and 3, IGMP snooping, and both PIM Sparse and Dense modes. Each version of IGMP introduced enhancements in multicast group management, from basic join/leave capabilities in IGMPv1 to source-specific filtering in IGMPv3. IGMP snooping was highlighted as a key feature in Layer 2 switches to prevent unnecessary multicast flooding. Additionally, the differences between PIM Sparse Mode optimized for widely dispersed multicast receivers and PIM Dense Mode ideal for tightly clustered groups were discussed in detail.

The theoretical concepts were reinforced through hands-on lab experiments, which demonstrated how multicast traffic behaves under various configurations. The lab allowed observation of real-time behavior of multicast routing tables, PIM joins and prunes, IGMP reports, and traffic flow transitions between the Root Path Tree and the Shortest Path Tree. These practical insights not only validated the theoretical knowledge but also deepened the understanding of how multicast operates in real-world network scenarios.

References

[l] S. Deering, "Host Extensions for IP Multicasting", STD 5, RFC 11 12, August 1989.

[2]Multicast Routing Algorithms and Protocols: A Tutorial, Laxman H. Sahasrabuddhe and Biswanath Mukherjee, University of California

[3] A. Chen, D. Lee, and P. Sinha, “Optimizing multicast performance in large-scale WLANs,” 27th International Conference on Distributed Computing Systems, IEEE, 2007.

[4] M. Christensen., K. Kimball, and F. Solensky, “Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches,” IETF RFC 4541, May 2006.

[5] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J.Padhye, R. Ramjee, and L. Ravindranath, “Dircast: A practical and efficient wi-fi multicast system," 17th

[6] B. Fermer, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", draftietf- pim-sm-v2-new-04.txt, Work in Progress, November 2001.

[7] C. Bot, S. Bhattacharyya, L. Gidiano, R. Rockell, J. Meylor, D. Meyer, G. Shepherd, B. Haberman, "An Overview of Source-Specific Multicast (SSM) Deployment", 'draft-ietf-ssm-overview-02.MW, Work in Progress, December 2001.