

What is IPv4 multicast?

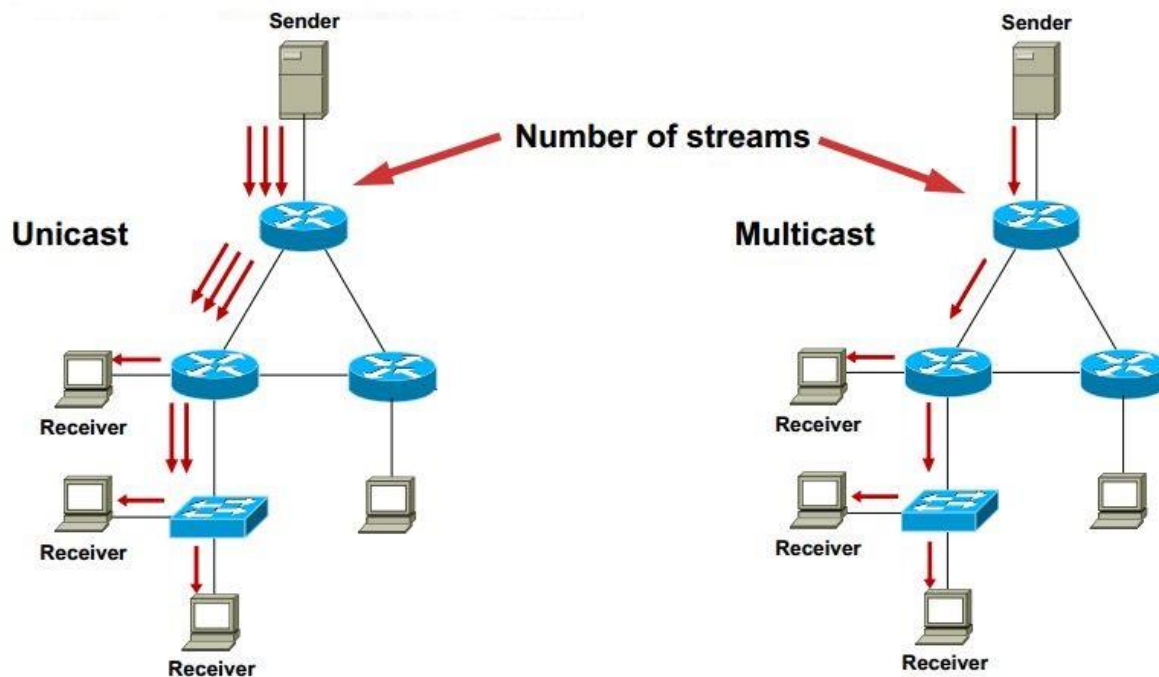
It is a bandwidth-conserving technology delivering a single stream of information simultaneously to a large or small number of endpoints within the network. such as video/audio streaming , Video conferencing ,corporate communications ,distance learning and software distribution.

In another meaning, IP multicast routing enables a source host to send packets to a group of receivers anywhere within the IP network by using a special form of IP address called the IP multicast group address. This special multicast group address forms the destination IP address of the packet.

Simply, multicast (use UDP) is the delivery of single data transmission to group of destinations on the same time. Source generates single data feed for all interested destinations (recipients) and does not need to know who they are. , routers make single forwarding decision for all recipients.

Deference between Unicast & Multicast

As you can see in the following diagram, in multicast only one flow of packets send which save the utilization of out bandwidth.

**Simulate Multicast Streaming using one Machine**

To feel the multicast we can use VLC media player to stream a video to many receivers.

We can test it assuming our machine is source and receiver on same time , just open two VLC instances (windows)and do same as the following:

Here's how to do it:

1. In the Media menu, choose "Stream"
2. In the Open Media dialog file tab, click "add" and choose the file you want to stream and click "Open"
3. At the bottom, click the "Stream" button

4. This opens the “Stream Output” dialog showing the source file you have chosen. Click Next to set destination.
5. In “Destinations”, choose “RTP /MPEG Transport Stream” and click the “Add” button
6. In the “Address” box, enter the required multicast address (eg 232.1.1.1) and set the port (or leave default at 5004)
7. In transcoding options, choose the appropriate settings for your video and PC’s codecs. I chose “Video H.264 + MP3 (MP4)”. I had to set the options by clicking the options (screwdriver and spanner) button immediately to the right of the dropdown. In encapsulation, I chose MPEG-TS. In video codec, I set the bitrate to 4000kb/s
8. Once the options are set, click “Save”. Then click Next for “Option Setup” and select “Stream all elementary streams” then click stream.

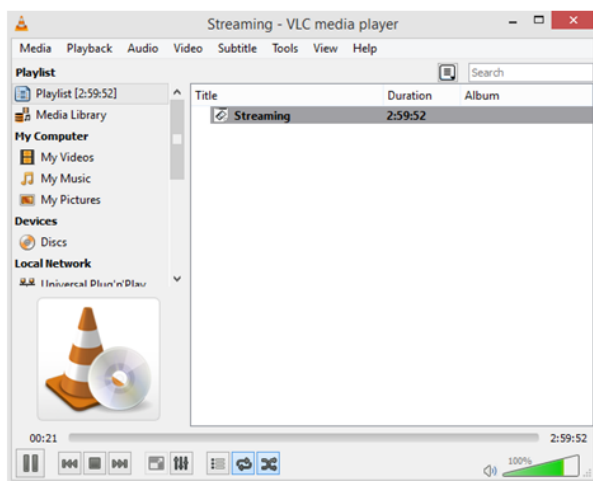
To view the stream, open another instance of VLC media player (try it on the same PC before trying it over the network)

1. Choose Media/Open Network Stream
2. In address, enter rtp://@232.1.1.1:5004 – choose the correct address and port you entered when setting up the stream. Don’t forget to enter the “@” symbol after “rtp://” and before the multicast ip address!
3. Click “Play”

If you want to stream multiple videos, remember to choose an different multicast address and/or port

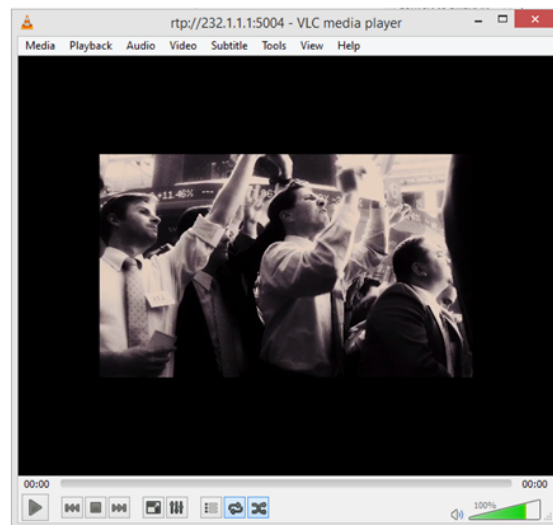
Sender (Source)

Send Video Streaming for a Movie
Using Multicast Group Address
232.1.1.1 port 5004



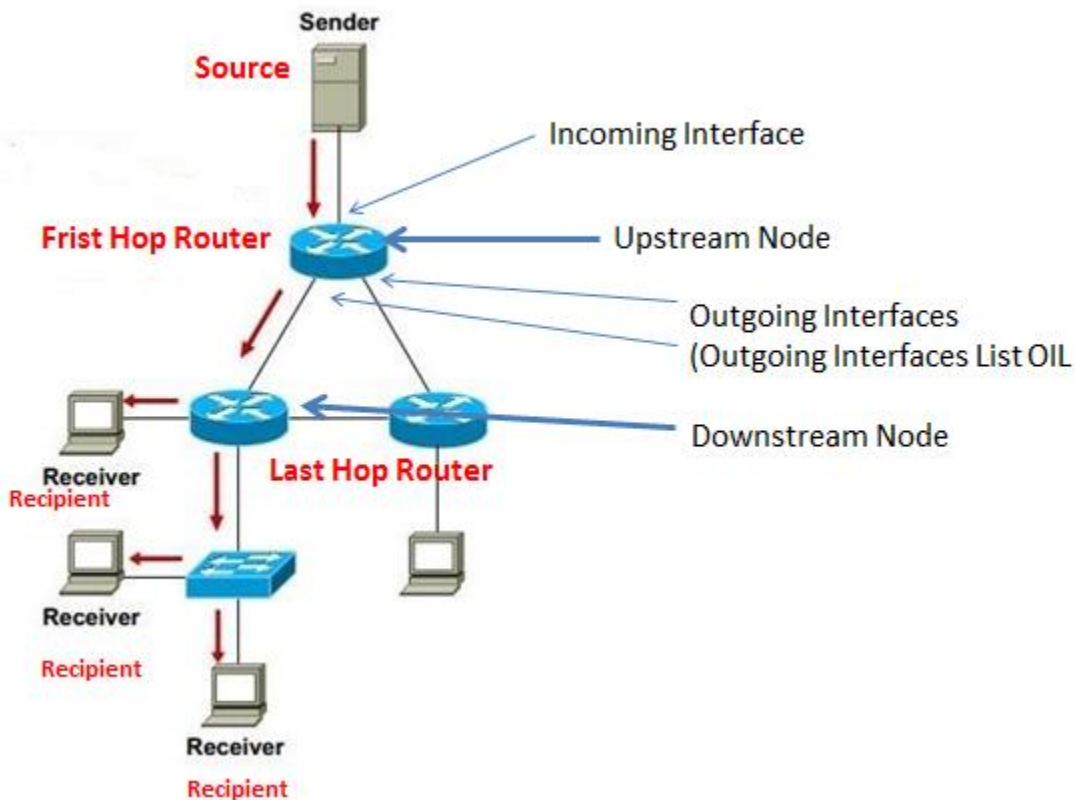
Receiver

Send Join Request to Group 232.1.1.1:5004
And start receiving Movie Streaming



Now the question, if we have routers and switches between sender & receivers so how we will configure them to forward this multicast traffic (Movie Streaming) ? To answer it we need to understand IPv4 Multicast Using Cisco Routers & Switches., **Let's Begin**

Multicast Topology Terms



Multicast Group IP address

Reserved Link –Local Address 224.0.0.0 to 224.0.0.255

Globally Scoped Address 224.0.1.0 to 238.255.255.255 for use on the internet

Source Specific Multicast 232.0.0.0 to 232.255.255.255 source specific delivery model

GLOP address 233.0.0.0 to 233.255.255.255 reserved for use on the internet by companies that process a publicly registered AS

Administrative Limited scope 239.0.0.0 to 239.255.255.255 reserved for internal corporate use

Multicast "group" address agreed between sender and receiver for specific feed

-source send traffic to destination address which is the group address

-receiver listen to traffic sent to group address

So Remember, Multicast address always in destination never used as source

How multicast work

1-source app send UDP multicast traffic with "group" destination address

2-interested receivers "join" group address by signaling routers

3-routers communicate to build loop free "tree" from sender to receiver

4-parts of the network without receivers will not receive traffic for that group

Protocols used in Multicast

Communications between Host & Router: IGMPv1, 2(default) and 3

Communications between Router to Router: PIM in IPv4, MLD in IPv6

Communications between Switch & Router: IGMP Snooping or CGMP

So we can say Multicast Main Components:

Control plane (routing) : IGMP , PIM

Data plane (forwarding) : RPF , Mroute table

Internet Group Management Protocol (IGMP)

How can hosts that are interested in receiving the multicast information actually join this multicast group?

Using Internet Group Management Protocol (IGMP) makes this possible in IPv4, while the Multicast Listener Discovery (MLD) protocol makes this possible in IPv6.

IGMP dynamically registers individual hosts in a multicast group on a particular LAN .once you enable PIM protocol.

When you enable PIM on interface, IGMP version 2 will be enabled automatically by default.

Simply , host talk to router with IGMP which is enabled automatically once we run PIM , pc send to router membership report (ask to be member in multicast address group)

IGMP Versions

IGMP version 1 :provides a basic query-response mechanism that allows the multicast router to determine which multicast groups are active; also enables hosts to join and leave a multicast group

It use two types to signal group membership:

host membership report used by client to join a group

host membership query used by router to see if members of group still exists

Simply , host when want to leave membership without telling router stop sending me multicast traffic. one of router is querier , receive requests from hosts and always send to 224.0.0.1 ask if any device want to request or be member in group , aged out 3 minutes if no answer he will no more hosts ask for requests, anyway version 1 is legacy & replaced with v2.

IGMP version 2: introduces the IGMP leave process, group specific queries, and an explicit maximum response time field; also adds the capability for routers to elect the IGMP querier without dependence on the multicast protocol to perform this task.

So we can say IGMP Version 2 enhance version 1 by adding :

querier election

if more than one routers on the lan who sends queries?

tunable timers

to speed up query response timeouts

group specific queries

query sent to the group address instead of all multicast hosts

explicit leave

speeds up convergence if no other hosts are joined to that group

Simply, host send leave message to router telling i don't want to be member any more on that group he send to 224.0.0.2 , in this version we have queries election to choose which router will be querier

IGMP version 3: provides source filtering; supports the link local address 224.0.0.22, which is the destination IP address for IGMP version 3 membership reports

Simply , Version 3 used to support SSM source specific multicast , use closed streaming server to host and send traffic to him.

Remember v1 not compatible with v2 while v2 compatible with v3

IGMP commands:

Int f0/0

ip igmp join-group 239.1.1.1

(will force Router to begin sending IGMPv2 membership reports , this command useful to test our multicast network by making receivers ping this multicast address and emulate this router connected to multicast streaming source)

Int f0/0

ip igmp version 1/2/3

(Change manually your IGMP version)

Multicast Distribution Trees (source trees and shared trees)

Source tree

Root is the source of the multicast traffic and whose branches form a spanning tree through the network to the receivers. Because this tree uses the shortest path through the network, it also is referred to frequently as a shortest path tree (SPT).

The special notation of (S, G), pronounced "S comma G", enumerates an SPT where S is the IP address of the source and G is the multicast group address. Using this notation, the SPT for the example (192.1.1.1, 224.1.1.1).

Where 192.1.1.1 is source of multicast streaming & 224.1.1.1 is the multicast group address used by receivers to reach this streaming

Simply, *Source Tree use shortest path from sender to receiver. Used in both sparse & dense mode*

Shared tree

Shared trees use a single common root placed at some chosen point in the network. Depending on the multicast routing protocol, this root is often called a rendezvous point (RP) or core, which lends itself to shared trees' other common names: RP trees (RPT) or core-based trees (CBT)

(* , 224.2.2.2) *means all sources, and the G represents the multicast group , which is in this example 224.2.2.2

Simply, *Shared Tree use shortest path from sender to Rendezvous Point (RP) then shortest path from RP to receiver. Used in sparse only.*

PIM protocol Independent Multicast

Used to build Router to Router Communication & to build loop-free "tree" from sender to receiver.

Simply, used to make router tell other routers he want specific multicast group G

Other protocols like PIM but legacy now: MOSPF, DVMRP

PIM does not maintain its own topology database. Instead, it relies on the unicast forwarding table of the router to avoid loops.

PIM had two versions: V1 , V2 and two modes Dense Mode , Sparse Mode

PIM Modes:

Dense push mode use source tree

Sparse pull mode use source tree or shared tree

PIM Dense Mode PIM-DM:

Flood multicast traffic for all groups out all multicast-enabled interfaces. Routers which determine they have no clients interested in receiving the traffic then send prune messages up toward the source, requesting that the flow of multicast traffic downstream be stemmed.

Simply, traffic ALWAYS send unless you do not want it.

PIM Sparse Mode PIM-SM:

Multicast traffic from a source isn't forwarded to group members. When a member somewhere in the network decides it wants to receive traffic for a group, it sends a join request to its nearest router. The join request is propagated up the multicast tree toward the source router. Upon receiving the join request, the source router begins forwarding multicast traffic for the group out the appropriate interface(s).

Simply, no traffic send unless you ask for it , this mode use Rendezvous Point RP to process join requests

Dense mode push router send to every one without asking hosts

Sparse mode pull igmp join send from host first

PIMv1 is a **Cisco propriety** protocol that can dynamically map RP's to multicast groups in concert

With a standalone1 protocol called **Auto-RP**. PIMv1 use time-to-live value to scope its announcements.

PIMv2 is a **standards**-based track protocol that made several improvements on the earlier Cisco proprietary version. These improvements include the concept of a single active RP per multicast group With multiple alternate RP's. In version 2,PIM packets are stand-alone packets and no longer embedded in IGMP messages. PIM v2 packets have support for automated fault tolerant RP discovery and distribution called a **Bootstrap router (BSR)**. This means that PIMv2 does not need any standalone protocols like Auto-RP does to allow routers to dynamically learn group-to-RP mappings.

Simply, remember that PIM had two Modes

PIM-Dense Mode (not required RP)

PIM-Sparse Mode (Require RP)

RP can be chosen statically by manually specify the RP address or can be chosen dynamically by using PIMv1 (Cisco Propriety) called Auto-RP or using PIMv2 (standards) called RP/BSR

Multicast Tables

Once tree from sender to receiver is built, traffic begin to flow ,when routers receive multicast packets , two tables will be checked to see how forwarding should occur:

1-CEF Table - called reverse path forwarding RPF check

2-Multicast Routing Table like unicast routing table controls what interfaces packets should forward to

Reverse Path Forwarding (RPF)

check source address and outgoing interface

in RPF check if incoming multicast interface == outgoing unicast int , RPF check passed

in RPF check if incoming multicast interface not equal outgoing unicast int , RPF check fails and packets dropped , which mean maybe loop would happen .

RPF checks are performed on each multicast packet received by a PIM router. RPF verifies that incoming multicast packets arrive on the interface closest to their source, by comparing the source address with the unicast routing table. We will talk later more about RPF

Multicast routing Table (mroute)

PIM monitoring the unicast routing table, also maintains its own multicast routing table to track incoming and outgoing interfaces for multicast traffic. Multicast routes are expressed in the format (S, G), where S is the multicast source and G is the group address.

Simply,

-routers know where sources and where receivers are

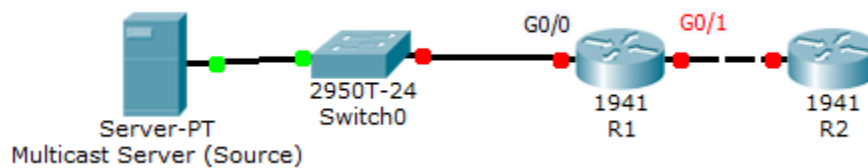
-interface facing downstream towards receivers make up "outgoing interface list " OIL

-split horizon like behavior stops an interface from being OIL if it's already incoming

-if RPF check passes , packets flow from incoming interface to all interfaces in the OIL

Example:

In R1 G0/0 is incoming , G0/1 is outgoing interfaces list (OIL)



We use `show ip mroute` to display multicast routing table

```
Router# 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,
      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
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:07:04/00:02:49, RP 2.2.2.2, flags: SJPL
  Incoming interface: FastEthernet0/0, RPF nbr 10.0.12.2
  Outgoing interface list: Null
```

PIM-DM

RFC 3973 , push model

All traffic flooded through LAN not centralized Router such as RP taking care of receivers requests also routers have no receivers prune (unjoin) the link , this mode had no scale because of flooding and (S,G) creation

1-Routers discover PIM neighbors using 224.0.0.13

2-routers once receive traffic insert (S,G) into routing table :

Incoming interface is attached to server , OIL is all other interfaces

3-flood all multicast traffic

4-prune unwanted traffic , Prune used to tell upstream neighbor to stop sending traffic for (S,G)

Prune occurs if :

- Multicast feed fails RPF check
- no downstream neighbors or receivers
- downstream neighbors already sent prune

Note when traffic flow stop , (S,G) remains in table .

PIM Dense Mode use some features to maintain Multicast routing Table :

Graft

What happen if i pruned (S,G) but then receive IGMP join message from receiver ?

Graft messages used in dense mode to unprune

Assert

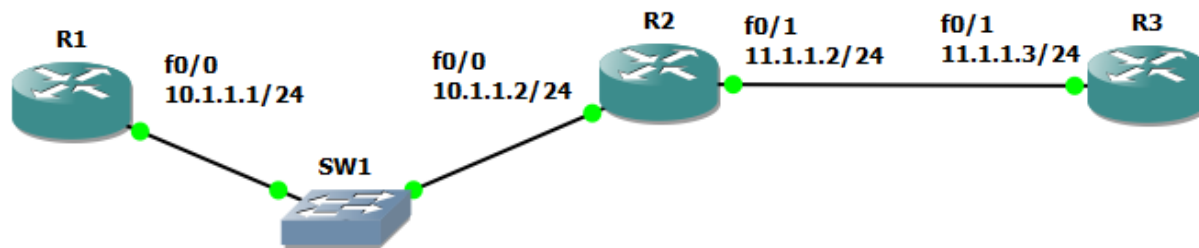
To prune duplicate multicast feed transmissions

-winner is lowest metric to source or if equal the highest ip address

State refresh

Once (S,G) is pruned , traffic re-flooded about every 3 minutes

So ,state refresh is keep alive for prune state

Configuring PIM-DM

Let's say R1 is connected to Multicast source and R3 will connect to Multicast Receivers .
 To simulate this source and test it in R1 we will let R1 f0/0 join multicast group 232.1.1.1.
 Remember all routers configured with RIP to provide layer 3 connectivity.

R1

```
ip multicast-routing
```

```
interface FastEthernet0/0
```

```
ip pim dense-mode
```

```
ip igmp join-group 232.1.1.1
```

R2

```
ip multicast-routing
```

```
interface FastEthernet0/0
```

```
ip pim dense-mode
```

```
interface FastEthernet0/1
```

```
ip pim dense-mode
```

R3

```
ip multicast-routing
```

```
interface FastEthernet0/1
```

```
ip pim dense-mode
```

To display R3 PIM interfaces: **show ip pim interfaces**

```
R3#sh ip pim int
```

| Address | Interface | Ver/Mode | Nbr Count | Query Intvl | DR Prior | DR |
|----------|-----------------|----------|-----------|-------------|----------|----------|
| 11.1.1.3 | FastEthernet0/1 | v2/D | 1 | 30 | 1 | 11.1.1.3 |

To display R3 PIM neighbors: **show ip pim neighbors**

```
R3#sh 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
```

| Neighbor Address | Interface | Uptime/Expires | Ver | DR Prio/Mode |
|------------------|-----------------|-------------------|-----|--------------|
| 11.1.1.2 | FastEthernet0/1 | 00:06:44/00:01:23 | v2 | 1 / S P G |

To display R1 int f0/0 groups joined: **show ip int f0/0 | I Multicast**

```
R1#sh ip int f0/0 | i Multicast
```

```
Multicast reserved groups joined: 224.0.0.1 224.0.0.2 224.0.0.22 224.0.0.13  

224.0.0.9
```

Notice Multicast Group address this interface joined is :

224.0.0.1 all devices

224.0.0.2 all hops(routers)

224.0.0.22 igmp v3

224.0.0.13 PIM

224.0.0.9 RIP

Display Multicast Routing Table : `show ip mroute`

```
R1#sh 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,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.1.1.1), 00:12:33/00:02:35, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  FastEthernet0/0, Forward/Dense, 00:12:33/stopped
```

To verify ping 232.1.1.1 from (Receiver) R3 :

```
R3#ping 232.1.1.1 rep 5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 232.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.1.1, 272 ms
Reply to request 1 from 10.1.1.1, 232 ms
Reply to request 2 from 10.1.1.1, 200 ms
Reply to request 3 from 10.1.1.1, 80 ms
Reply to request 4 from 10.1.1.1, 80 ms
^C
```

Or

Ping using extended ping so you can specify R3 source Interface will be used on this ping

```

R3#ping
Protocol [ip]:
Target IP address: 232.1.1.1
Repeat count [1]: 5
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: fastethernet 0/1
% Invalid interface

Interface [All]: fastethernet0/1
Time to live [255]:
Source address or interface: 11.1.1.3
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 232.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 11.1.1.3

Reply to request 0 from 10.1.1.1, 96 ms
Reply to request 1 from 10.1.1.1, 228 ms
Reply to request 2 from 10.1.1.1, 240 ms
Reply to request 3 from 10.1.1.1, 84 ms
Reply to request 4 from 10.1.1.1, 204 ms

```

```

R1#sh 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,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.1.1.1), 00:13:57/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:13:57/stopped

(11.1.1.3, 232.1.1.1), 00:00:32/00:02:27, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.1.2
  Outgoing interface list: Null

R1#

```

```

R2#sh 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,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.1.1.1), 00:16:08/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/1, Forward/Dense, 00:16:06/stopped
    FastEthernet0/0, Forward/Dense, 00:16:08/stopped

(11.1.1.3, 232.1.1.1), 00:02:47/00:00:12, flags: T
  Incoming interface: FastEthernet0/1, RPF nbr 11.1.1.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:02:47/stopped

```

```

R1#sh ip mroute count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
1 routes using 554 bytes of memory
1 groups, 0.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 232.1.1.1, Source count: 0, Packets forwarded: 0, Packets received: 0
R1#

```

PIM-SM

RFC 4601, use PULL model or "explicit join"

Uses both shared trees & source Based trees while dense mode use only source tree.

How sparse mode shared tree work?

- 1-discover PIM neighbor & elect DR
- 2-discover RP
- 3-tell RP about sources & receivers
- 4-build shared tree from sender to receiver through RP
- 5-join shortest path tree
- 6-leave shared tree

Remember, two trees in shared tree:

- From receiver to RP
- From RP to sender (source)

What is RP ?

RP is used as reference point for the root of the shared tree

RP learns about sources through unicast PIM register messages tells about (S,G)

RP learns about receivers through PIM join messages tells to add an interface to the OIL

RP is used to merge the two trees together

Register messages (way to know sources):

When first hop routers connected to source hears traffic , a unicast register message sent to RP
(If many first hop router exists, the DR registers)

If RP accept this message it ack with register Stop and insert (S,G) into table

At this point only DR and RP know (S,G)

Join Message (way to know receivers):

When last hop router receive IGMP join , PIM join is generated up the reverse path tree towards the RP

All routers in the reverse path install (*.G) AND FORWARD THE JOIN HOP-BY-HOP to the RP

At this point RP and all downstream devices towards the receivers know (*.G)

Merging the trees

Once RP know about sender & receiver for same group , he will send PIM join message up to source

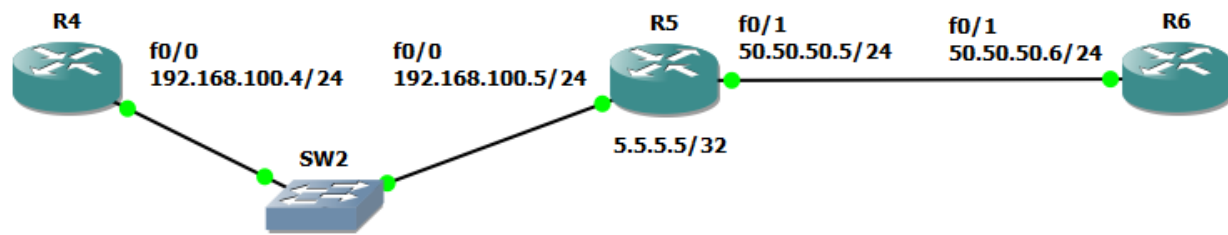
All routers in the path from RP to source will install (*.G) with OIL point to RP

Like dense mode, sparse mode use state refresh to ensure that feeds do not timeout

PIM-Sparse Mode (Require RP)

RP can be chosen statically by manually specify the RP address or can be chosen dynamically by using PIMv1 (Cisco Propriety) called Auto-RP or using PIMv2 (standards) called RP/BSR

Every Multicast address must serve by one RP address

Configuring PIM-SM Static RP**R4**

```

ip multicast-routing
interface FastEthernet0/0
ip pim sparse-mode
ip igmp join-group 232.2.2.2

```

ip pim rp-address 5.5.5.5 < Must type it on all routers including RP it self's

R5

```

ip multicast-routing
interface FastEthernet0/0
ip pim sparse-mode
interface FastEthernet0/1
ip pim sparse-mode

```

ip pim rp-address 5.5.5.5

R6

```

ip multicast-routing
interface FastEthernet0/1
ip pim sparse-mode

```

ip pim rp-address 5.5.5.5

```

R6#sh ip pim int

Address          Interface          Ver/  Nbr   Query  DR    DR
                  Mode              Count Intvl Prior   DR
50.50.50.6       FastEthernet0/1   v2/S    1    30     1     50.50.50.6
R6#sh ip pim nei
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface          Uptime/Expires    Ver    DR
Address                               Prio/Mode
50.50.50.5         FastEthernet0/1    00:11:59/00:01:35 v2      1 / S
R6#

```

To display RP address on any router we type: **show ip pim rp mapping**

```
R6#sh ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
RP: 5.5.5.5 (?)
R6#
```

```
R4#sh 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,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.2.2.2), 00:13:18/00:02:27, RP 5.5.5.5, flags: SJPL
Incoming interface: FastEthernet0/0, RPF nbr 192.168.100.5
Outgoing interface list: Null

(*, 224.0.1.40), 00:13:18/00:02:24, RP 5.5.5.5, flags: SJPL
Incoming interface: FastEthernet0/0, RPF nbr 192.168.100.5
Outgoing interface list: Null
```

```
R5#sh 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,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.2.2.2), 00:14:12/00:02:42, RP 5.5.5.5, flags: SJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
FastEthernet0/0, Forward/Sparse, 00:14:12/00:02:42

(*, 224.0.1.40), 00:14:56/00:02:54, RP 5.5.5.5, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
FastEthernet0/1, Forward/Sparse, 00:14:00/00:02:52
FastEthernet0/0, Forward/Sparse, 00:14:58/00:02:46
```



```
R6#ping 232.2.2.2 rep 5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 232.2.2.2, timeout is 2 seconds:

Reply to request 0 from 192.168.100.4, 284 ms
Reply to request 1 from 192.168.100.4, 64 ms
Reply to request 2 from 192.168.100.4, 212 ms
Reply to request 3 from 192.168.100.4, 308 ms
Reply to request 4 from 192.168.100.4, 40 ms
R6#
```

```
R4#sh 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,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.2.2.2), 00:16:10/stopped, RP 5.5.5.5, flags: SJPL
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.100.5
  Outgoing interface list: Null

(50.50.50.6, 232.2.2.2), 00:00:47/00:02:43, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.100.5
  Outgoing interface list: Null

(*, 224.0.1.40), 00:16:10/00:02:29, RP 5.5.5.5, flags: SJPL
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.100.5
  Outgoing interface list: Null
```

```

R5#sh 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,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 232.2.2.2), 00:15:55/stopped, RP 5.5.5.5, flags: SJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:15:55/00:02:05

(50.50.50.6, 232.2.2.2), 00:01:17/00:02:11, flags: JT
  Incoming interface: FastEthernet0/1, RPF nbr 50.50.50.6
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:01:17/00:02:05

(*, 224.0.1.40), 00:16:41/00:03:05, RP 5.5.5.5, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/1, Forward/Sparse, 00:15:45/00:03:05
    FastEthernet0/0, Forward/Sparse, 00:16:41/00:01:58

```

Notes:

- **sparse-dense mode.** This means that we will use sparse mode if possible, if there is no RP we will use dense mode. We will take later about using **ip pim sparse-dense mode**
- To configure router as RP write on it and all other routers :
ip pim rp-address
- we can use RP for specific address and another for other address , for instance in the following commands we assign RP 1.1.1.1 for 224.1.1.1 , 224.1.1.2 , 224.1.1.3 , RP 2.2.2.2 for 224.2.2.2 , 224.2.2.3 , 224.2.2.4

```

ip pim RP-address 1.1.1.1 2
ip pim RP-address 2.2.2.2 3

```

```

access-list 2 permit 224.1.1.1
access-list 2 permit 224.1.1.2
access-list 2 permit 224.1.1.3
access-list 3 permit 224.2.2.2
access-list 3 permit 224.2.2.3
access-list 3 permit 224.2.2.4

```

- when an access-list created for one RP you need to create another access-list for other RP since RP without any ACL will always by default serve all multicast groups (224.0.0.0 15.255.255.255)
- when write ACL you should write on all multicast domain routers including RP
- if two RP have overlapping scope of groups , RP with higher address wins.

What is PIM DR?

When we configure PIM on our routers we will establish PIM neighbor adjacencies and the PIM hello messages are also used to elect a designated router for each multi-access network. The DR is the router that will forward the PIM join message from the receiver to the RP (rendezvous point).

Since the DR is used to forward PIM join messages to the RP , it's not used with PIM-Dense Mode where no RP exists .

We always will see DR in Multi-access Network (Switch in use) .

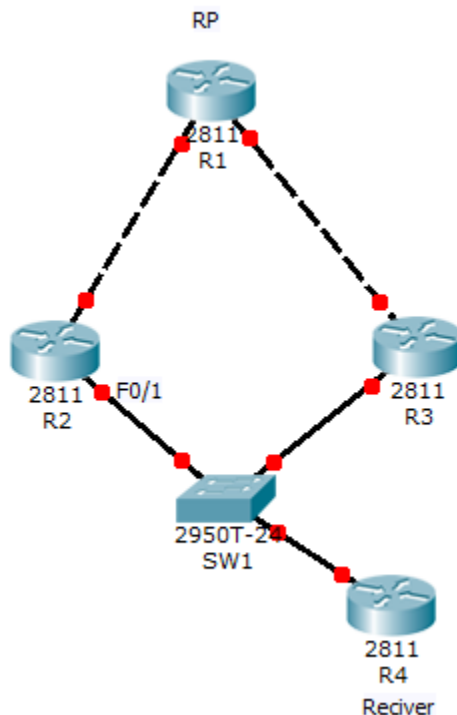
The router with Highest DR Priority is chosen as DR (default is 1) if two routers had same default Priority then router with highest IP address is chosen as the DR.

To change Router DR Priority :

Int f0/0

Ip pim dr-pri 2

Let's see the following example



Here we would had problem since when R4 RP receive join message from both R2&R3 and send Multicast streaming it will be send to R2&R3 where both will send same (duplicate) stream , to avoid this problem we need only R2 OR R3 become DR (responsible for forward PIM join messages to our RP) Let's say we want R2 to become DR :

Int f0/1

Ip pim dr-pri 100

Unlike OSPF there is no BDR (Backup Designated Router) in PIM. When the DR fails, other routers will see this because their PIM neighbor adjacency will go down. A new election will take place and another router will become the DR.

PIM-SM Dynamic using RP Auto-RP (Cisco propriety PIM Version1)

Another way to configure RP is Auto-RP (cisco Proprietary method to announce RP-mapping)
Auto-RP requires that you configure the RPs to announce their availability as RPs to mapping agents. The RPs use **224.0.1.39** to send their announcements.

RP mapping agent listens to the announced packets from the RPs, then sends RP-to-group mappings in a discovery message that is sent to **224.0.1.40**.

So we had Two Roles in Auto-RP

- **Candidate RP** (Device Advertise his willing to be RP using 224.0.1.39)
- **Mapping Agent MA** (Device choose RP among candidates and announce it to the rest of PIM domain using 224.0.1.40) telling other routers which group-to-RP mapping to use.

You can configure both roles on same router.

If there are more than One RP announce itself for given Group, Mapping agent will elect the RP with the highest Ip address as RP for this Multicast Group address

How AUTO-RP work?

1-Candidate RP generate advertisement using (S,224.0.1.39) with groups they are willing to service .

2-Mapping agent listens for (*,224.0.1.39) to learn about Candidate RPs.

Then generates advertisement using (S,224.0.1.40) to distribute RP mapping

The chicken & Egg issue

Routers must join (*,224.0.1.39) for candidate RP & (*,224.0.0.40) for Mapping agent BUT 224.0.1.39 & 224.0.1.40 used by PIM-SM RP candidates & Mapping agent work only In Dense Mode

To solve this we use of two solutions:

1-use **ip pim sparse-dense** mode under interfaces instead of ip pim sparse , This means that we will use sparse mode if possible, if there is no RP we will use dense mode.

2-on all routers on your PIM Domain including RPs & Mapping agents we type **Ip Pim autorp listener** when we type this command on all routers in topology , it change 224.0.1.39 , 40 to be used with dense mode and the rest of groups will use Sparse Mode (so we can know our RP).

Remember if you configured your network with static RP and Auro-RP on same time the following rule applied "notes:

Dynamically learned RP mappings are preferred over statically configured ones

To configure router as Auto-RP only in this router write:

ip pim send-rp-ann loop0 scope 5 interval 10

loop0 mean I will announce myself as router willing to be RP using my loop0 address BUT you must enable PIM-SM or PIM-SM-DM on the loopback interface first .

Scope is ttl for packet used to define boundary for your network.

Interval every 30 sec i send advert

To configure router as Auto-RP Mapping Agent write only in this router:

```
ip pim send-rp-dis loop0 scope 5
```

We can make RP candidate router announce its self as RP for specific multicast group address

```
access-list 1 permit 238.0.0.0 0.255.255.255
```

```
ip pim send-rp-announce loop0 scope 2 group-list 1
```

Mapping Agent should be protected against candidate RP so **RP announce filter** feature can permit or deny RP to be accepted.

In mapping agent we type :

```
access-list 1 permit host 1.1.1.2
```

```
access-list 2 deny any
```

```
ip pim rp-announce-filter rp-list 1 group-list 2
```

rp-list use acl point to 1.1.1.2 , that's mean do not elect the router with this address as RP , anything else will not subject to this filter.

group-list use acl point to deny all networks , this mean we will not select 1.1.1.2 as RP for all multicast group address .

To filter router from our pim neighbor which we man we do not want to have PIM neighbor relationship with him :

```
access-list 1 deny 10.1.13.1
```

```
int fas 0/0
```

```
ip pim neighbor-filter 1
```

sometime we would like to keep the PIM neighbor relationship exists but we want the Router to not send RP announcements to his neighbor .

This what we call **IP Multicast Boundary**

```
int f0/0
```

```
ip multicast boundary 1
```

```
access-list 1 deny host 224.0.1.40
```

```
access-list 1 permit any
```

Note: PIM-DM will be used if we typed "ip pim sparse-dense-mode" for groups with no RP, we can disable this behavior using "**no ip pim dm-fallback**"

PIM-SM Dynamic using RP Bootstrap Router BSR (Standards PIM version2)

If all routers in the network are running PIMv2, you can configure a BSR instead of Auto-RP. BSR and Auto-RP are very similar. A BSR configuration requires that you configure BSR candidates (similar to RP-Announce in Auto-RP) and BSRs (similar to Auto-RP Mapping Agents).

We had two roles

- **RP Candidates**, using unicast PIM to advertise itself to BSR
- **Bootstrap Router BSR act as Mapping Agent**

In RP Candidates we type:

```
ip pim rp-can loop0
```

In Bootstrap router we type:

```
ip pim bsr-can loop0
```

Router can be rp can and bsr on same time

by default auto-rp & bsr messages sent to all PIM enabled interfaces , for security these messages should be filtered on network edge ,so autorp use multicast boundary But bsr use BSR Border , also we can filter using scope keyword.

If you have more than one router act as RP-candidates, the one with higher ip address will be elected by BSR as RP unless we change the RP priority

```
ip pim rp-candidate loop0 priority 200
```

(Default is zero , higher priority preferred)

To display BSR router address we type :

```
Sh ip pim bsr-router
```

Rate Limit

We can configure RPs to rate limit the number of register messages for instance to 10/sec
This will sets a limit on the maximum number of PIM-SM register messages sent per second for each (S,G) entry which is useful to limit the load on DR & RP , message exceeded this threshold will be dropped.

```
ip pim register-rate-limit 10
```

TTL-threshold

By default any locally generated Multicast packets sent with TTL of 255
Only multicast packets with TTL greater than the interface TTL threshold value are forwarded on the interface (by default set to zero which mean all multicast packets are forwarded)

```
ip multicast ttl-threshold 252
```

Debug IPv4 Multicast

If you want to debug multicast traffic you have to disable multicast route caching on the required interface

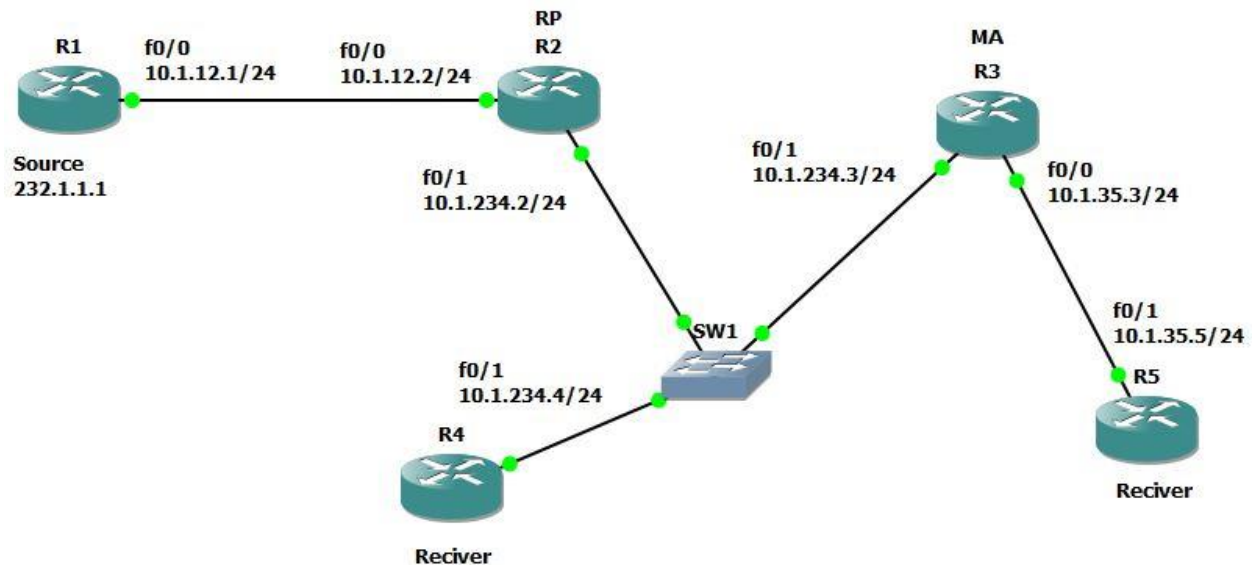
```
R3 (config) #interface serial 0/0
```

```
R3 (config-if) #no ip mroute-cache
```

Then type

```
R3#debug ip mpacket
```

Configuring Auto-RP & BSR Lab



R1

```
ip multicast-r
int f0/0
ip add 10.1.12.1 255.255.255.0
no sh
ip pim sparse-m
ip igmp join-group 232.1.1.1
router rip
ver 2
no au
net 0.0.0.0
```

R2

```
ip multicast-r
int f0/0
ip add 10.1.12.2 255.255.255.0
no sh
ip pim sparse-m
int f0/1
ip add 10.1.234.2 255.255.255.0
no sh
ip pim sparse-m
```



```
int loop 0
ip add 2.2.2.2 255.255.255.255
ip pim sparse-m
router rip
ver 2
no au
net 0.0.0.0
```

R3

```
ip multicast-r
int f0/0
ip add 10.1.35.3 255.255.255.0
no sh
ip pim sparse-m
int f0/1
ip add 10.1.234.3 255.255.255.0
no sh
ip pim sparse-m
int loop 0
ip add 3.3.3.3 255.255.255.255
ip pim sparse-m
router rip
ver 2
no au
net 0.0.0.0
```

R4

```
ip multicast-r
int f0/1
ip add 10.1.234.4 255.255.255.0
no sh
ip pim sparse-m
router rip
ver 2
no au
net 0.0.0.0
```

R5

```
ip multicast-r
int f0/0
ip add 10.1.35.5 255.255.255.0
no sh
ip pim sparse-m
router rip
ver 2
no au
net 0.0.0.0
```

To configure Auto-RP where R2 RP & R3 MA
R2

```
ip pim send-rp-ann loop0 scope 5 interval 10
```

R3

```
ip pim send-rp-dis loop0 scope 5 interval 10
```

on all routers

```
ip pim autorp lis
```

R5#ping 232.1.1.1 rep 5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 232.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 96 ms

Reply to request 1 from 10.1.12.1, 96 ms

Reply to request 2 from 10.1.12.1, 88 ms

Reply to request 3 from 10.1.12.1, 100 ms

Reply to request 4 from 10.1.12.1, 112 ms

To configure BSR where R2 RP & R3 BSR

Remove ip pim autorp lis and send-rp-ann , send-rp-dis we wrote above for auto-rp , then type the following :

R2

```
ip pim rp-can loop0
```

R3

```
ip pim bsr-can loop0
```

Lest verify our BSR , clear ip mroute 232.1.1.1 then

```

R5#sh ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 2.2.2.2 (?), v2
    Info source: 3.3.3.3 (?), via bootstrap, priority 0, holdtime 150
    Uptime: 00:03:09, expires: 00:02:15
R5#ping 232.1.1.1 rep 5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 232.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 96 ms
Reply to request 1 from 10.1.12.1, 124 ms
Reply to request 2 from 10.1.12.1, 84 ms
Reply to request 3 from 10.1.12.1, 84 ms
Reply to request 4 from 10.1.12.1, 112 ms
R5#mtra
R5#mtrace 232.1.1.1
% Bad source address
R5#mtrace 10.1.12.1
Type escape sequence to abort.
Mtrace from 10.1.12.1 to 10.1.35.5 via RPF
From source (?) to destination (?)
Querying full reverse path...
 0 10.1.35.5
-1 10.1.35.5 PIM [10.1.12.0/24]
-2 10.1.35.3 PIM [10.1.12.0/24]
-3 10.1.234.2 PIM [10.1.12.0/24]
-4 10.1.12.1

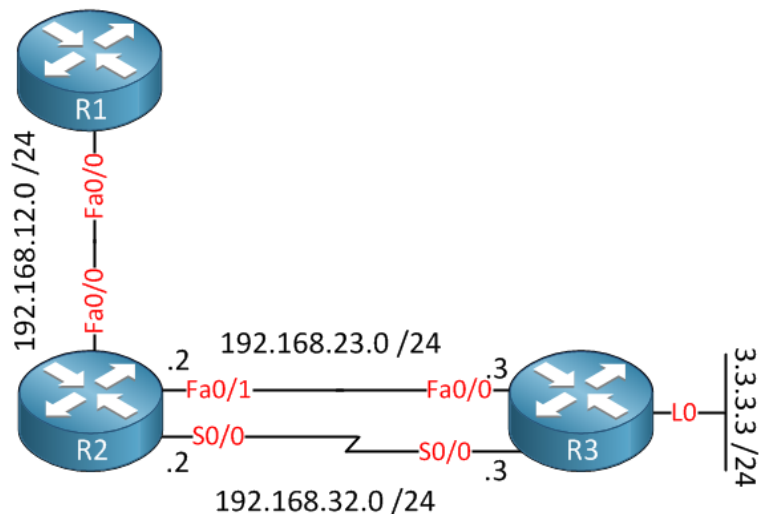
```

RPF & why sometimes we use static or default multicast route (ip mroute) command.

PIM uses the unicast routing table to check what interface will be used to reach the source.

PIM will only accept multicast packets on an interface that we use to reach the source. If we receive multicast packets on an interface that we don't use to reach the source, we will drop the multicast packets! This is called a RPF failure.

Let's check this topology which is running OSPF by the way.



What if we want to use serial link between R2 & R3 for multicast But we enable PIM-DM on R2 serial only and in R3 we enable PIM-DM on fastethernet link

If we type `show ip mroute 239.1.1.1` in R2 we will notice outgoing interface is s0/0 but R3 using fastethernet link, in this case we can go to R3 and type

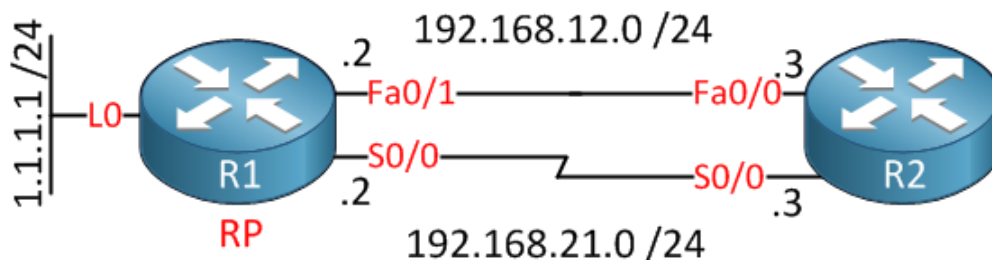
```
R3(config)#ip mroute 192.168.12.1 255.255.255.255 serial 0/0
```

Now lets check R3 outgoing interface :

```
R3#show ip mroute 239.1.1.1
```

```
(*, 239.1.1.1), 00:11:17/stopped, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Loopback0, Forward/Dense, 00:11:17/00:00:00
Serial0/0, Forward/Dense, 00:11:17/00:00:00
```

RPF failures apply to the **data plane AND control plane**. In the example above you saw an example of a RPF failure on the data plane. Lets check RPF failure on the control plane:



OSPF is running and R1 as the RP (Rendezvous point).

PIM-SM enabled on R2 serial interface

But OSPF will choose Fastethernet as preferred link since it had better cost but

```
R2#show ip mroute 239.1.1.1
```

```
(*, 239.1.1.1), 00:01:16/00:01:43, RP 1.1.1.1, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Serial0/0, Forward/Sparse, 00:01:16/00:01:43
```

Above you see that R2 tries to join the RP for this multicast group, but if you take a close look you can see that there is no incoming interface and that the RPF neighbor is 0.0.0.0.

The problem is that R2 will do a RPF check how it can reach the RP, it should be the serial link but the unicast routing table is telling us something different:

```
R2#show ip route ospf
```

```
1.0.0.0/32 is subnetted, 1 subnets
O 1.1.1.1 [110/11] via 192.168.12.1, 00:09:26, FastEthernet0/0
```

If we want to fix this, we need to create a static multicast route to solve the RPF error:

```
R2(config)#ip mroute 1.1.1.1 255.255.255.255 serial 0/0
```

So , Why use ip mroute ?

One of the key differences between unicast and multicast is that for unicast routing we only care about where the destination is located and how to get there. For multicast routing we care about where the source is located. PIM (Protocol Independent Multicast) uses the unicast routing table to check what interface will be used to reach the source.

PIM will only accept multicast packets on an interface that we use to reach the source. If we receive multicast packets on an interface that we don't use to reach the source, we will drop the multicast packets! This is called a RPF failure and it's the #1 issues why multicast isn't working for many networking students.

We can use

```
R3(config)#ip mroute 192.168.12.1 255.255.255.255 serial 0/0
```

to avoid mismatching interfaces when we receive multicast traffic from int but out RPF neighbor is diff int and as a result multicast traffic will be dropped.

PIM-SSM (Source Specific Multicast)

The multicast we knew (PIM sparse and dense mode) using IGMPv2 are also known as ASM (Any Source Multicast). Which means that the receivers really don't care what source they receive multicast traffic from, all sources are accepted.

SSM (Source Specific Multicast) RFC 4607 requires IGMPv3 and lets us join multicast groups from specified source addresses.

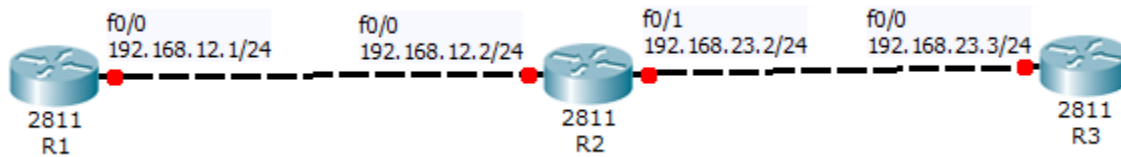
It identifies a set of multicast hosts not only by group address but also by source. so the receiver is able to receive the group from a specified source. An SSM group called a channel. IANA has reserved for SSM the IPv4 address range **232.0.0.0/8**.

When using SSM / IGMPv3 there are no shared trees. We only build SPTs (Shortest Path Trees) towards our sources.

We don't use any RPs ...(no need for Auto-RP or Bootstrap anymore)

SSM brings several important benefits over ASM. Because an SSM channel is defined by both a source and a group address, group addresses can be re-used by multiple sources while keeping channels unique. For instance, the SSM channel (192.168.45.7, 232.7.8.9) is different than (192.168.3.104, 232.7.8.9), and hosts subscribed to one will not receive traffic from the other.

Configuring source specific multicast SSM is easy, let's use the following topology to take a look at the configuration:



Run ip pim sparse-mode on all interfaces

on all routers enable ssm

Router(config)# **ip pim ssm default**

This will enable PIM-SSM for its default group range (232.0.0.0/8)

Remember in SSM only the router which is closest to the receiving host (last hop router) needs to have SSM enabled .

An access list can be specified in place of the default range to enable PIM-SSM for a different range if desired. For instance

On all routers :

Access-list 1 permit 232.0.0.0 0.255.255.255

ip pim ssm range 1

Enable IGMPv3 on the source and receiver

R1(config)#interface fastEthernet 0/0

R1(config-if)#**ip igmp version 3**

R3(config)#interface fastEthernet 0/0

R3(config-if)#**ip igmp version 3**

Now let's configure R3 to join a multicast group address in the 232.0.0.0/8 range and we'll specify R1 as the source

R3(config)#interface fastEthernet 0/0

R3(config-if)#**ip igmp join-group 232.1.1.1 source 192.168.12.1**

R2#show ip mroute 232.1.1.1

(192.168.12.1, 232.1.1.1), 00:00:18/00:03:10, flags: sT

Incoming interface: FastEthernet0/0, RPF nbr 192.168.12.1

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:00:18/00:03:10

Multicast Helper-Map

In financial trading networks where a legacy stock ticker application sends packets out as broadcast UDP so we need to convert broadcast to multicast, The router on the attached segment can then convert the broadcast destination to multicast, send the packet into the multicast transit network, and then on the last hop router attached to the receiver translate the multicast packet back to a broadcast.

Multicast helper-map command is similar in theory to how the unicast “ip helper-map” works. With the IP helper map feature, IP broadcast packets, such as UDP based DHCP requests, have their destination addresses translated to a unicast address, such as the DHCP server. With the IP multicast helper map feature, IP broadcast packets have their destination addresses translated to a multicast address.

Example:

SW1 — R4 — R3 — R2 — R1 — SW2

SW1 is the broadcast sender (i.e. the source application), SW2 is the receiver (i.e. the destination application), R4 is the first hop router, and R1 is the last hop router. IGP and PIM adjacencies exist between R4 – R3, R3 – R2, and R2 – R1.

R4’s configuration, the first hop router, looks as follows:

```
R4#
interface FastEthernet0/0
description TO SENDER APPLICATION – SW1
ip address 173.20.47.4 255.255.255.0
ip multicast helper-map broadcast 224.1.2.3 100
!
ip forward-protocol udp 31337
access-list 100 permit udp any any eq 31337
```

This configuration means that if R4 receives a UDP broadcast going to port 31337 inbound on Fa0/0 it will be translated to the multicast address 224.1.2.3. Note that the use of the “ip forward-protocol” command is necessary in order to process switch UDP traffic going to the port in question. Without process switching the helper-map feature can not correctly categorize and translate the traffic.

R1’s configuration, the last hop router, looks as follows:

```
R1#
interface Serial0/0
description TO R2
ip address 173.20.12.1 255.255.255.0
ip pim dense-mode
ip multicast helper-map 224.1.2.3 173.20.18.255 100
!
interface FastEthernet0/0
description TO RECEIVER – SW2
ip address 173.20.18.1 255.255.255.0
ip directed-broadcast
```



```
ip forward-protocol udp 31337
access-list 100 permit udp any any eq 31337
```

This configuration means that if R1 receives a UDP multicast going to the group address 224.1.2.3 at port 31337 inbound on S0/0.102 it will be translated to the directed broadcast address 173.20.18.255. Since the link 173.20.18.0/24 is directly connected and has the directed broadcast address of 173.20.18.255 by default, the configuration implies that traffic matching the helper map on S0/0.102 will be sent as a broadcast out Fa0/0.

Note the use of the “ip forward-protocol” command as before in order to process switch the UDP traffic. Additionally the “ip directed-broadcast” command is enabled on the last hop outgoing interface since in current IOS versions this is disabled by default for security purposes.

Bidirectional PIM

sparse mode had two trees (shared & source tree)
unidirectional SPT from source to RP
unidirectional SPT from RP to Receivers

The downside of using PIM sparse mode with many active sources and receivers is that we will see many mroute state entries that can take quite some resources. Using PIM sparse mode the RP builds 2 entries:

(*,G)
(S,G)

Results will see a lot of (*,G) & (S,G) entries in control plane

For many to many multicast application this will not be scale well
so Bidirectional PIM has been invented for networks where we have many sources and receivers talking to each other. An example of this is videoconferencing where it's not just one source with many listeners but all the participants are communicating with each other.

Use Bidirectional PIM by only allowing the shared tree (*,G) and never a SPT (S,G)

When we use PIM bidirectional mode the RP will never build a (S,G) entry and we only allow the (*,G) entry for the shared tree. PIM routers will never build the SPT (Shortest Path Tree) towards the source.

Another difference between PIM sparse mode and PIM bidirectional mode is that with sparse mode traffic only flows down the shared tree. Using PIM bidir mode traffic will flow up and down the shared tree! Also PIM bidirectional does not use the PIM register / register-stop mechanism to register sources to the RP. Each source is able to start sending to the source whenever they want.

When the multicast packets arrive at the RP they will be forwarded down the shared tree (if there are receivers) or dropped (when we don't have receivers). There is however no way for the RP to tell the source to stop sending multicast traffic.

Design-wise you really have to think about where to place the RP in your network as it should be somewhere in the middle between the sources and receivers in the network.

Last but not least...PIM bidirectional has no RPF check. There is a different solution to prevent loops, we will use a **DF (Designated Forwarder)**. This designated forwarder is the only router on the segment that is allowed to send multicast traffic towards the RP. When there is only 1 router per segment that forwards multicast traffic there will be no loops.

The DF will be elected using the following mechanism:

The router with the lowest metric to the RP will be the DF.

If the metric is equal then the router with the highest IP address will become the DF.

Simply, *one DF is elected per PIM segment*

Similar to Assert, lowest metric to the RP wins or Highest ip address if metrics was the same.

Only DF can forward traffic upstream towards RP.

All other interfaces in OIL are downstream facing

Bid PIM Removes the need for RPF check but all routers must agree on Bidir or loop can occur

How Bid PIM works?

1-defines RP and group range as bidirectional stops (S,G) for that range

2-build signal (*,G) TREE towards RP where:

- traffic flows upstream from source to RP
- traffic flows downstream from RP to receivers

3-remove PIM register process

Traffic from sources always flows to RP

4-use Designated forwarder DF for Loop prevention

BID PIM Configuration

Let's say we want routers support multicast forwarding for groups in two directions.

On all interfaces: ip pim sparse-m

Enable PIM Bidirectional **on all PIM-enabled routers: ip pim bidir-enable**

On all routers configure RP address :

ip pim rp-address 1.1.1.1 bidir

Note that I added the "bidir" keyword. This is required to allow Router with 1.1.1.1 address to be the RP for PIM bidirectional. To keep things simple I'm statically configuring the RP. You can also use AutoRP or BSR (Bootstrap) just don't forget to add the "bidir" keyword.

R3#**show ip pim rp mapping**

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static, Bidir Mode

RP: 1.1.1.1 (?)

R2#**show ip pim interface df**

* implies this system is the DF

| Interface | RP | DF Winner | Metric | Uptime |
|-----------------|---------|---------------|--------|----------|
| FastEthernet0/0 | 1.1.1.1 | *192.168.24.2 | 11 | 00:01:36 |
| FastEthernet0/1 | 1.1.1.1 | 0.0.0.0 | 11 | 01:05:08 |

Above you see which router is the DF for each segment. The * indicates that this router is the DF for the given link.

DF is multicast router that is able to forward (*,G) state in 2 different directions for same group address. If two routers in the path, DF winner is determined by IGP cost on the link.

Unlike PIM-SM normal operations, RP does not need to be in that data path but just must be reachable by all multicast routers.

Finally, if you routers Bidirectional state is not created for unnecessary groups

On all routers:

Access-list 1 deny 224.0.1.40

Access-list 1 permit 224.0.0.0 15.255.255.255

Ip pim rp-address 1.1.1.1 1 bidir

This access-list will allow the routers to allow Bidirectional forwarding for all groups except the 224.0.1.40

Multicast Source Discovery Protocol (MSDP)

MSDP is your solution when implementing IP multicasting across different Autonomous Systems (ASs). Each AS still uses Protocol Independent Mode Sparse Mode (PIM-SM) protocol to build the data distribution tree within the domain by using a Rendezvous Point (RP). MSDP will operate between the RPs in each AS, enabling them to discover multicast sources active in other ASs.

- 1- When a multicast source is registered with the RP using normal PIM-SM procedures
- 2- When configured for MSDP, the RP sends Source Active (SA) messages to RP in another AS. The SA messages contain information about the source and the multicast group to which it is sending traffic.
- 3- The SA messages received by the RP are also forwarded to other configured RPs that are downstream.
- 4- The originating RP continues to send periodic SA messages for the (S,G) every 60 seconds as long as the source is sending packets to the group.

RFC3618

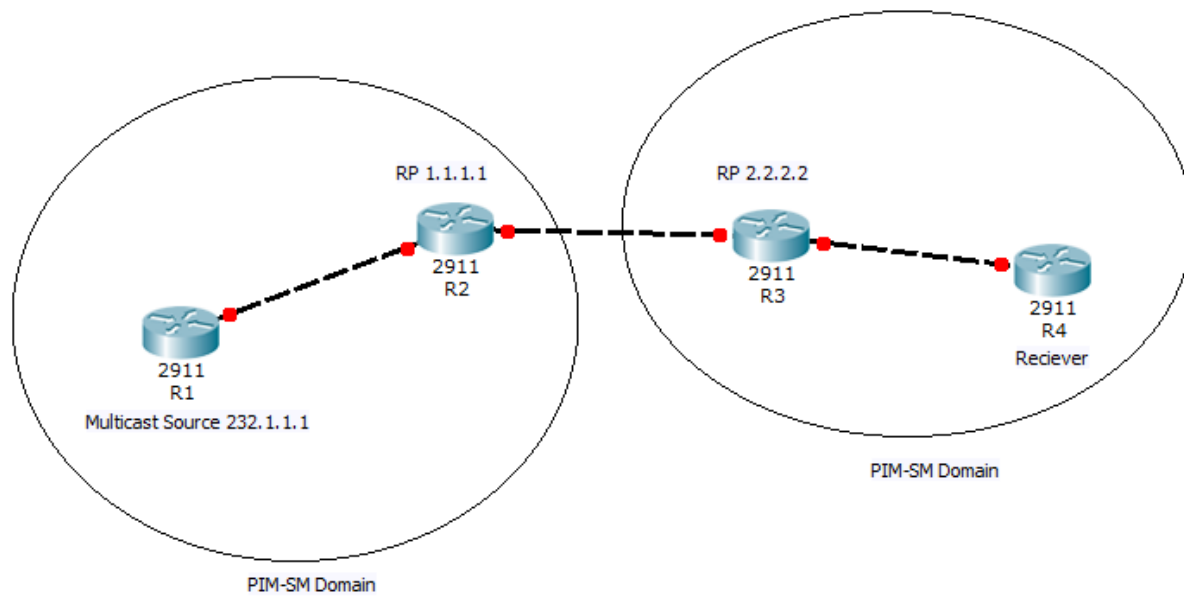
MSDP tells RPs: who is the active multicast source

MSDP uses unicast TCP

When RP receives PIM register for (S,G) = active sources, he sends copy to msdp peer using Source Active SA message.

Other RP in the another AS takes this SA and caches it in (msdp-sa-cache) for TTL but will not do this by default.

Other RP will not add (S,G) he got from another RP to multicast routing table unless his receivers start asking to join this source.

In which Scenarios MSDP will be used?**Scenario 1**

Normal Use for MSDP , we could have two different PIM domain each with its own RP , and one PIM domain had Receivers and the other domain had the Multicast Source .

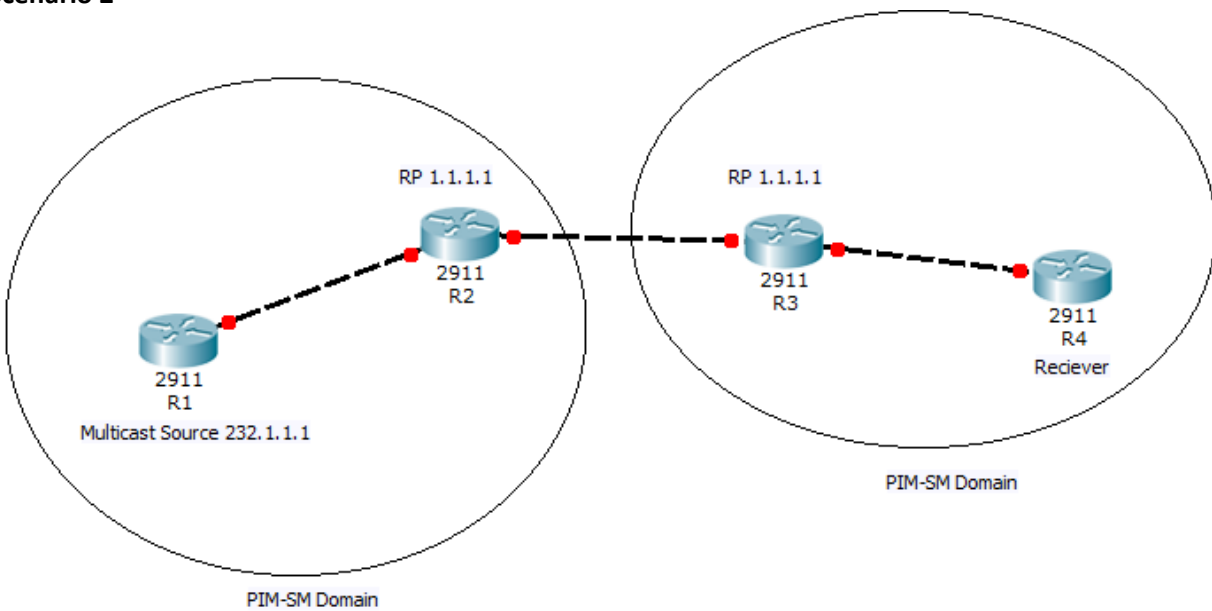
Remember Only One RP per single Multicast Group Address.

R2

ip msdp peer 10.1.23.3 connect-source loop 0

R3

ip msdp peer 10.1.23.2 connect-source loop 0

Scenario 2

We call this scenario **Any Cast** where RP address is the same used for two separated PIM-SM Domain

So What is any cast ?

- There are a number of method to create redundancy for our RPs (Rendezvous Points) in our multicast topology. Using AutoRP or Bootstrap we can configure multiple routers to advertise themselves as RPs (Rendezvous Points) so when one router fails, another one can take over. Any cast RP is a different method to create redundancy...
- To make sure that all Rendezvous points know about all the different sources out there we will use MSDP (Multicast Source Discovery Protocol). Normally MSDP is used to allow multicast routing between different autonomous systems (BGP) but it's also a great companion for any cast RP.
- *Traffic from source to destination reaches closest destination in the topology*
- Any Cast useful when multicast goes on internet and we have many sources and we want closet source to your region will send you the stream
- All RPs must share the same information about senders and receivers
- MSDP will make sure that PIM register is sent to one any cast RP but PIM join will not sent to another.
- Any Cast RP are MSDP peers using unique address (globally routed address) , if we have more than two RPS , we use mesh group.

How its work?

- 1-anycast RP1 and RP2 peer with msdp
- 2-source S sends traffic to group G
- 3-PIM DR registers (S,G) with RP1

- 4-RP1 send msdp SA to RP2
- 5-recivers send IGMP join for (*.G)
- 6-last hop router sends (*,G) PIM join towards RP2
- 7-RP2 send (S,G) join to S , tree is built

R2

ip msdp peer 10.1.23.3 connect-source loop 0

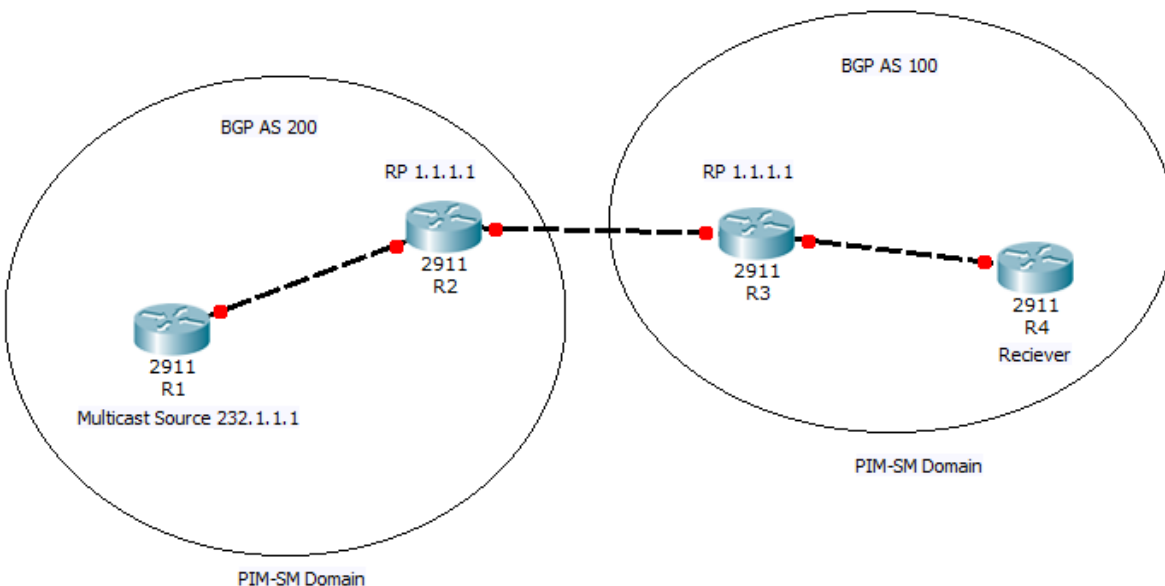
R3

ip msdp peer 10.1.23.2 connect-source loop 0

Note : **ip msdp sa-limit 10.1.1.1 40**

Limit the number of SA it receives from 10.1.1.1 to 40 SAs's

Scenario 3



MSDP MB-BGP

Here we have each PIM domain on separated BGP AS , let think about it in more details

For multicast to travel through internet all hops must run multicast but what if RPF check for multicast source is via unicast only peer ? Multicast BGP solves this by separating unicast RPF and multicast RPF

ip msdp peer 10.1.23.3 connect-source loop 0 remote-as 100

ip msdp originator-id Loopback 1

(The ip msdp originator-id command is used to tell MSDP what IP address to use as an "ID" which is similar to the OSPF or BGP router ID. By default the router will select the highest IP address on the router which means that R2 and R3 both would have selected 23.23.23.23.)

ip msdp description

Just a description you can write for specified peer

ip msdp cahce-sa-state

This let your router cache SA =(S,G) received , also we can let him cache SA from peers pass ACL lets say

acl number 10**ip msdp cache-sa-state list 10****ip msdp sa-request 1.1.1.1**

Instead of waiting SA to come in periodic times , RP can send request and get immediate response for all active sources for a given group , so once new source added your 1.1.1.1 which is your MSDP peer will send it immediately to you, we can add acl to filter request will received

ip msdp filter-sa-request 1.1.1.1 list 10

If we just type ip msdp filter-sa-request without acl , all SA will be filtered from that peer

Remember:

- MSDP: is mechanism to connect multiple PIM-SM domains enable multiple sources for a given group to be known for all RPs in the same or different domains.
- MSDP must be configured peering between RPs TCP 639
- RP send unicast register message (encapsulated mcast flow) to other RP which in his turn will de-encapsulated and send to last hop router
- If MSDP configured on the rp , packet also re-encapsulated in source active SA message and forward to all MSDP peers
- Creates SA state (cache source/group pairs). Those pairs that pass the access list are cached we use : **ip msdp cache-sa-state list 100 (100 acl #)**

Filtering MSDP

By creating an MSDP filter, you can do one of the following:

- Filter all source/group pairs
- Specify an extended access list to pass only certain source/group pairs
- Filter based on match criteria in a route map

To apply an MSDP filter, use the following commands in global configuration mode as needed:

| Command | Purpose |
|--|---|
| Router(config)# ip msdp sa-filter out { <i>peer-address</i> <i>peer-name</i> } | Filters all SA messages to the specified MSDP peer. |
| Router(config)# ip msdp sa-filter out { <i>peer--address</i> <i>peer-name</i> } list <i>access-list</i> | To the specified MSDP peer, passes only those SA messages that pass the extended access list. |
| Router(config)# ip msdp sa-filter out { <i>peer-address</i> <i>peer-name</i> } route-map <i>map-name</i> | To the specified MSDP peer, passes only those SA messages that meet the match criteria in the route map <i>map-tag</i> value. |

Controlling Source Information That Your Router Receives

By default, the router receives all SA messages its MSDP RPF peers send to it. However, you can control the source information you receive from MSDP peers by filtering incoming SA messages. In other words, you can configure the router not to accept them.

You can do one of the following to control the source information you receive from MSDP peers:

- Filter all incoming SA messages from an MSDP peer
- Specify an extended access list to pass certain source/group pairs
- Filter based on match criteria in a route map

To apply a filter, use the following commands in global configuration mode as needed:

| Command | Purpose |
|---|--|
| Router(config)# ip msdp sa-filter in { <i>peer-address</i> <i>peer-name</i> } | From the specified MSDP peer, filters all SA messages received. |
| Router(config)# ip msdp sa-filter in { <i>peer-address</i> <i>peer-name</i> } list <i>access-list</i> | From the specified MSDP peer, passes incoming SA messages that pass the extended access list. |
| Router(config)# ip msdp sa-filter in { <i>peer-address</i> <i>peer-name</i> } route-map <i>map-name</i> | From the specified MSDP peer, passes only those SA messages that meet the match criteria in the route map <i>map-name</i> value. |

Other issues we need to understand about MSDP for CCIEv5 R&S:

MSDP Mesh Group

When your domain had multiple RPs and you had group of msdp peers fully meshed (connected) to each other's , even in mesh msdp our SA received by peer will not forwarded to other peers in same mesh group.

To create msdp mesh group named Cbtme

On all mspd peers we type : **ip msdp mesg-group Cbtme**

Shutdown MSDP Peer

For temporary shutdown one of your MSDP peer we type

ip msdp shutdown 1.1.1.1

Monitoring MSDP**Sh ip msdp sum****Sh ip msdp count** (display numbers of sources cached)**Sh ip msdp peer****Sh ip msdp sa-cache** (display S,G state learned from msdp peer)**Debug ip msdp** (debug msdp activity)**Debug ip msdp resets** (debug msdp peer reset reasons)**Clear ip msdp peers** (clear tcp session with peer and resting all msdp messages counters)**Clear ip msdp statistics****Clear ip msdp sa-cache** (clears SA cache entries for all entries all sources for specific group)**Multicast over GRE tunnel interfaces**

Nothing much to say just remember this , internet (unicast networks) will not connect multicast domains unless we use tunnels.

*Tunnel interface treated like normal PIM interface**We could have possible issues with RPF failure and we use static multicast routing to resolve this*

For more information about Multicast over GRE : <http://www.cisco.com/c/en/us/support/docs/ip/ip-multicast/43584-mcast-over-gre.html>

Multicast Labs**PIM-DM Lab**

All routers running RIPv2:

router rip

version 2

network 0.0.0.0

no auto-summary

ip multicast-routing

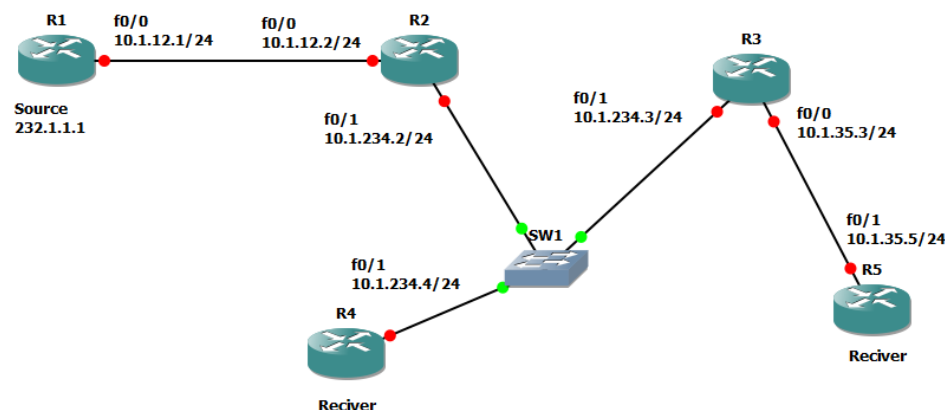
All routers interfaces PIM enabled:

ip pim dense-mode

R1

interface FastEthernet0/0

ip igmp join-group 232.1.1.1



Verifications :

```

sh ip mroute
sh ip rpf 10.1.12.1
ping 232.1.1.1 rep 5
  
```

PIM-SM Static RP

All routers running RIPv2

All routers interfaces PIM enabled:

ip pim sparse-mode

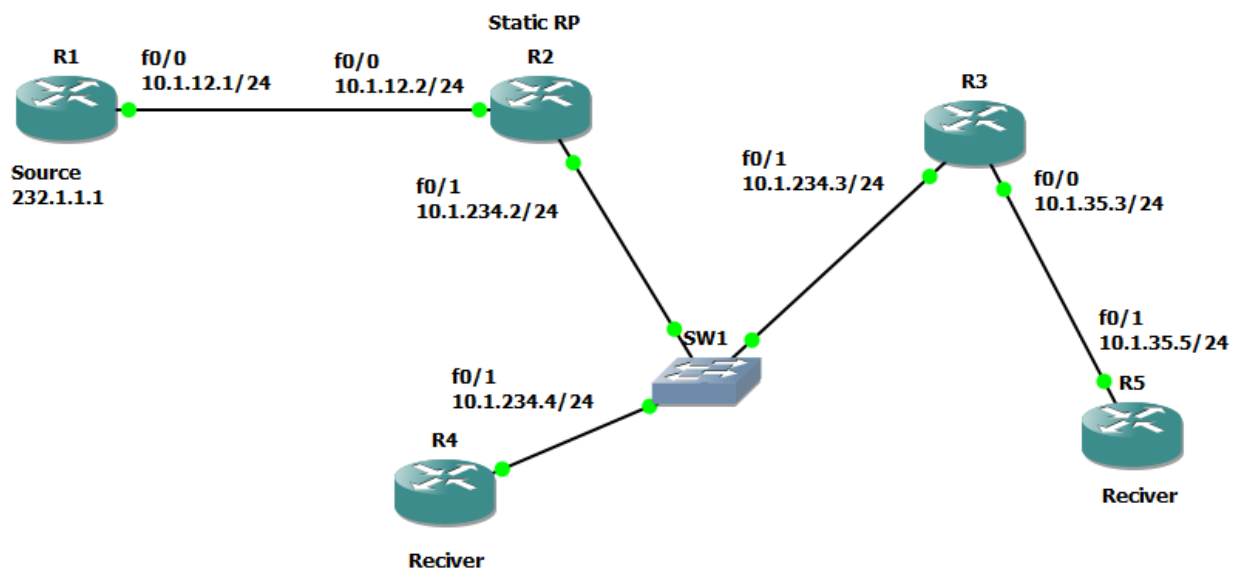
All routers configured to use RP:

ip pim rp-address 2.2.2.2

R1

interface FastEthernet0/0

ip igmp join-group 232.1.1.1



R5#ping 232.1.1.1

Reply to request 0 from 10.1.12.1, 156 ms

Notice R2 RP before ping had two (*,G)

R2#sh ip mroute

(*, 232.1.1.1)

(*, 224.0.1.40)

Notice R2 RP after ping and one (S,G)

R2#sh ip mroute

(*, 232.1.1.1)

(*, 224.0.1.40)

(10.1.35.5, 232.1.1.1)

PIM-SM Auto-RP

All routers running RIPv2:

All routers interfaces PIM enabled:

ip pim sparse-mode

R2 RP Candidate :

ip pim send-rp-announce Loopback0 scope 5 interval 10

R3 MA Mapping Agent :

ip pim send-rp-discovery Loopback0 scope 5 interval 10

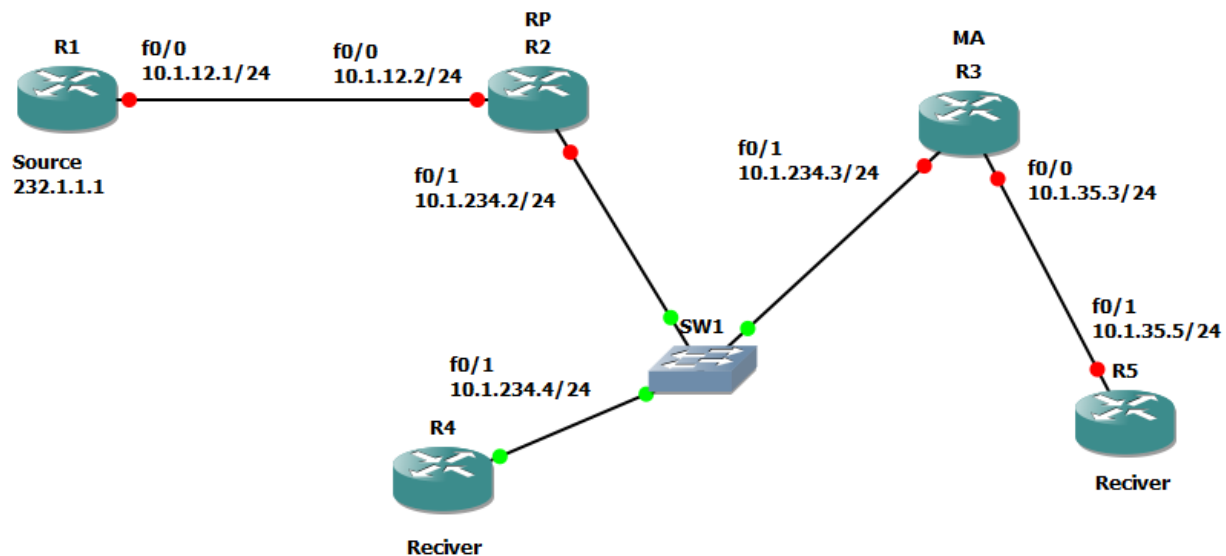
All routers :

ip pim autorp listener

R1

interface FastEthernet0/0

ip igmp join-group 232.1.1.1



R5#**sh ip pim rp mapping**

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4

RP 2.2.2.2 (?), v2v1

Info source: 3.3.3.3 (?), elected via Auto-RP

Uptime: 00:01:41, expires: 00:00:22

R5(config)#ip host R2 2.2.2.2

R5#sh ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
RP 2.2.2.2 (R2), v2v1
Info source: 3.3.3.3 (?), elected via Auto-RP
Uptime: 00:02:46, expires: 00:00:30

R2#sh ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 224.0.0.0/4
RP 2.2.2.2 (?), v2v1
Info source: 3.3.3.3 (?), elected via Auto-RP
Uptime: 00:03:38, expires: 00:00:30

R3#sh ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)

Group(s) 224.0.0.0/4
RP 2.2.2.2 (?), v2v1
Info source: 2.2.2.2 (?), elected via Auto-RP
Uptime: 00:04:20, expires: 00:00:29

After ping 232.1.1.1 from R5

R2#sh ip mroute

(* , 232.1.1.1)
(10.1.35.5, 232.1.1.1)
(* , 224.0.1.39)
(2.2.2.2, 224.0.1.39)
(* , 224.0.1.40)
(3.3.3.3, 224.0.1.40)

PIM-SM Bootstrap BSR

All routers running RIPv2:

All routers interfaces PIM enabled:

ip pim sparse-mode

R2 RP Candidate :

ip pim rp-candidate Loopback0

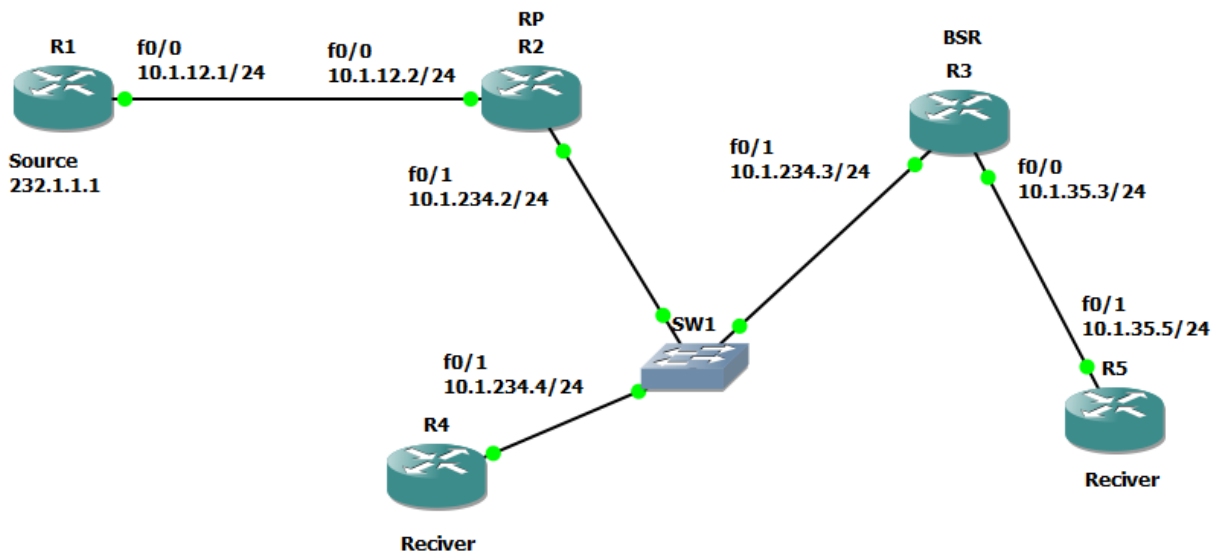
R3 BSR Mapping Agent :

ip pim bsr-candidate Loopback0 0

R1

interface FastEthernet0/0

ip igmp join-group 232.1.1.1



R2#**sh ip pim bsr-router**

PIMv2 Bootstrap information

BSR address: 3.3.3.3 (?)

Uptime: 00:01:41, BSR Priority: 0, Hash mask length: 0

Expires: 00:01:28

Candidate RP: 2.2.2.2(Loopback0)

Holdtime 150 seconds

Advertisement interval 60 seconds

Next advertisement in 00:00:06

```
R3#sh ip pim bsr-router
```

```
PIMv2 Bootstrap information
```

This system is the Bootstrap Router (BSR)

```
BSR address: 3.3.3.3 (?)
```

```
Uptime: 00:02:59, BSR Priority: 0, Hash mask length: 0
```

```
Next bootstrap message in 00:00:01
```

```
R2#sh ip pim rp mapping
```

```
PIM Group-to-RP Mappings
```

This system is a candidate RP (v2)

```
Group(s) 224.0.0.0/4
```

```
RP 2.2.2.2 (?), v2
```

```
Info source: 3.3.3.3 (?), via bootstrap, priority 0, holdtime 150
```

```
Uptime: 00:00:25, expires: 00:02:13
```

```
R3#sh ip pim rp mapping
```

```
PIM Group-to-RP Mappings
```

This system is the Bootstrap Router (v2)

```
Group(s) 224.0.0.0/4
```

```
RP 2.2.2.2 (?), v2
```

```
Info source: 10.1.234.2 (?), via bootstrap, priority 0, holdtime 150
```

```
Uptime: 00:00:41, expires: 00:01:47
```

```
R5#sh ip pim bsr-router
```

```
PIMv2 Bootstrap information
```

```
BSR address: 3.3.3.3 (?)
```

```
Uptime: 00:02:46, BSR Priority: 0, Hash mask length: 0
```

```
Expires: 00:01:24
```

```
R5#sh ip pim rp mapping
```

```
PIM Group-to-RP Mappings
```

```
Group(s) 224.0.0.0/4
```

```
RP 2.2.2.2 (?), v2
```

```
Info source: 3.3.3.3 (?), via bootstrap, priority 0, holdtime 150
```

```
Uptime: 00:01:04, expires: 00:01:33
```

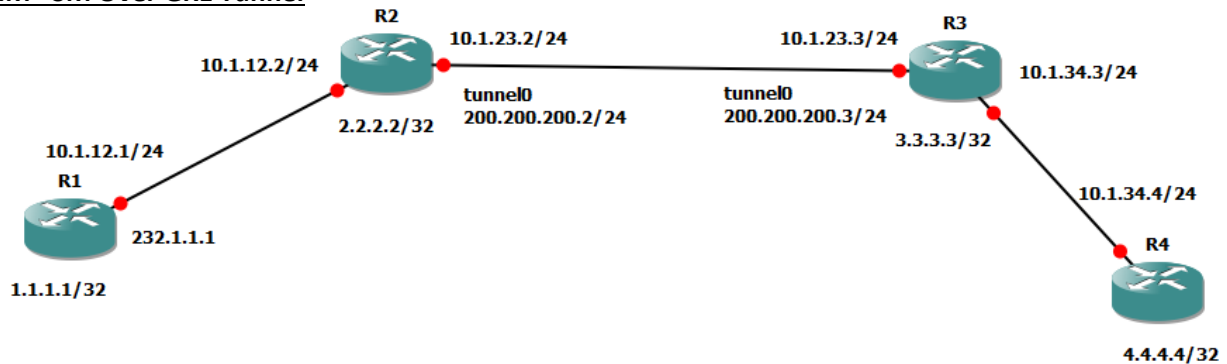
After ping

```
R2#sh ip mroute
```

```
(*, 232.1.1.1)
```

```
(10.1.35.5, 232.1.1.1)
```

```
(*, 224.0.1.40)
```

PIM –SM Over GRE Tunnel

All routers running : ospf 100 area 0

On all interfaces fastethernet, loops, tunnel:

ip pim sparse-m

on all routers :

ip pim rp-add 2.2.2.2

We advertise 200.200.200.0/24 in ospf but not 10.1.23.0/24

OR

advertise 10.1.23.0/24 in ospf but not 200.200.200.0/24 but also we

R2(config)#ip mroute 10.1.34.4 255.255.255.255 tunnel0

R3(config)#ip mroute 10.1.12.1 255.255.255.255 tunnel0

R4#ping 232.1.1.1 rep 5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 232.1.1.1, timeout is 2 seconds:

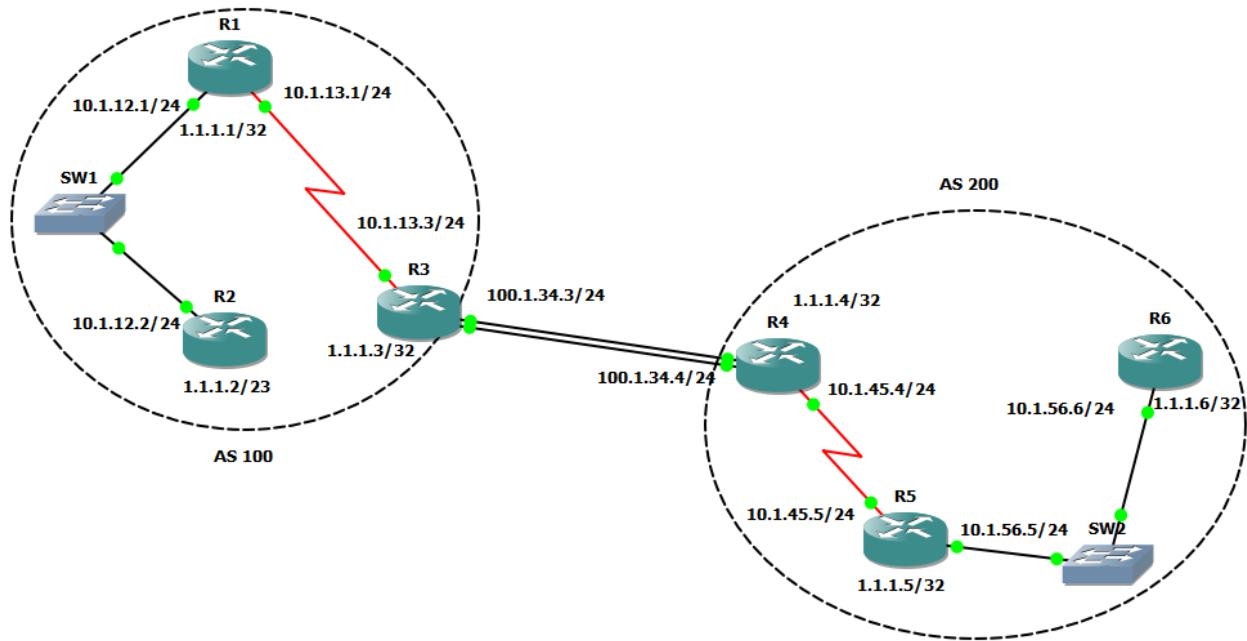
Reply to request 0 from 10.1.12.1, 132 ms

Reply to request 1 from 10.1.12.1, 84 ms

Reply to request 2 from 10.1.12.1, 84 ms

Reply to request 3 from 10.1.12.1, 84 ms

Reply to request 4 from 10.1.12.1, 84 ms

MSDP - MP-BGP**Running IGP****On R1,R2,R5,R6**

```
router ospf 100
net 0.0.0.0 0.0.0.0 are 0
```

no ospf between R3&R4**R3**

```
router ospf 100
net 1.1.1.3 0.0.0.0 are 0
net 10.1.13.3 0.0.0.0 are 0
```

R4

```
router ospf 100
net 1.1.1.4 0.0.0.0 area 0
net 10.1.45.4 0.0.0.0 area 0
```

on all routers

```
int loop 0
ip ospf network point-to-point
```

All routers interfaces PIM enabled:

```
ip pim sparse-mode
```

Configuring MP-BGP & Redis**R3**

```
router bgp 100
no bgp def ipv4-uni
nei 100.1.34.4 remote-as 200
add ipv4 uni
nei 100.1.34.4 act
net 1.1.1.1 mask 255.255.255.255
```

```
router ospf 100
default-info originate always
```

R4

```
router bgp 200
no bgp def ipv4-uni
nei 100.1.34.3 remote-as 100
add ipv4 uni
nei 100.1.34.3 act
net 1.1.1.5 mask 255.255.255.255
```

```
router ospf 100
default-info originate always
```

R1 will be RP for AS 100**R5 will be RP for AS 200**

```
R1,R2,R3
ip pim rp-add 1.1.1.1
```

```
R4,R5,R6
ip pim rp-add 1.1.1.5
```

R1 MSDP P R5

```
R1
ip msdp peer 1.1.1.5 connect-source loop0 remote-as 200
R5
ip msdp peer 1.1.1.1 connect-source loop0 remote-as 100
```

```
R5#sh ip msdp summ
MSDP Peer Status Summary
Peer Address   AS   State  Uptime/  Reset SA  Peer Name
              Downtime Count Count
1.1.1.1       100  Up     00:03:27 1    0    ?
```

R2 source

```
int f0/0  
ip igmp join-group 224.1.2.3
```

ping will not work in R6 since it's not advertised

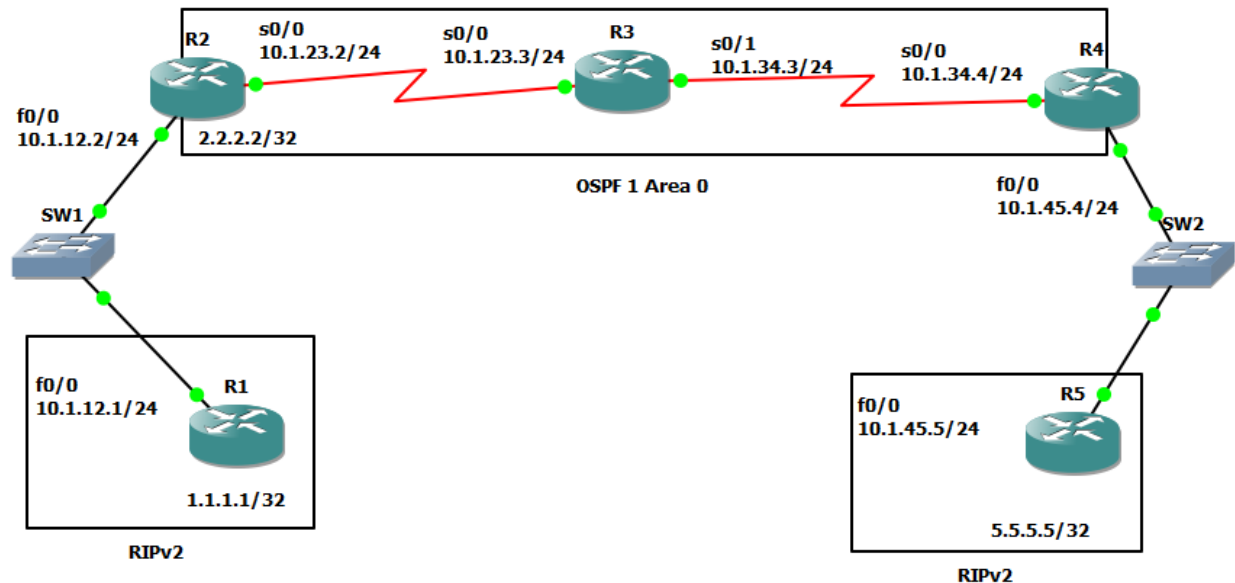
```
R4  
router bgp 200  
add ipv4  
net 1.1.1.6 mask 255.255.255.255
```

```
R6#ping 224.1.2.3
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:

```
Reply to request 0 from 10.1.12.2, 208 ms  
R6#
```

Multicast Helper Map

R1
 int f0/0
 ip rip v2-broadcast

R1 configured to send RIPv2 updates with broadcast

R2
 ip multicast-r
 int f0/0
 ip pim sparse-m
 int s0/0
 ip pim sparse-m
 int loop0
 ip pim sparse-m

ip pim rp-can loop0
 ip pim bsr-can loop0

R3
 ip multicast-r
 int s0/0
 ip pim sparse-m
 int s0/1
 ip pim sparse-m

R4
 ip multicast-r
 int s0/0
 ip pim sparse-m

ACL is configured to identify RIP traffic being sent from 10.1.12.1 to a broadcast ip address destined for RIP

R2

```
access-list 100 permit udp host 10.1.12.1 eq rip host 255.255.255.255 eq rip
```

now we need to specify the forwarding of broadcast messages destined to UDP 520 (RIP)

R2

```
ip forward-protocol udp rip
```

now we need to convert broadcast traffic arriving at f0/0 (interface of First hop router = router closest to source) destined for UDP port 520 to multicast group destination address 224.1.1.1

R2

```
int f0/0
```

```
ip multicast helper-map broadcast 224.1.1.1 100 ttl 3
```

ttl here for broadcast packets will be converted and send by R2 to reach R5

range is 1-50 hops

normally RIP use ttl2 but this mean it will reach r4 only so we made it 3

on R4

```
no access-list 100 permit udp host 10.1.12.1 any eq rip
```

```
ip forward-protocol udp rip
```

```
int s0/0
```

```
ip multicast helper-map 224.1.1.1 10.1.45.255 100
```

```
int f0/0
```

```
ip directed-broadcast
```

R5

```
router rip
```

```
no validate-update-source
```

R1

```
clear ip route *
```

R5

```
sh ip route rip
```

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

```
R    1.1.1.1 [120/1] via 10.1.12.1, 00:00:04
```

Resources:

Multicast Topics from René - CCIE #41726

<http://networklessons.com/category/multicast/>

Multicast Topics from Packetlife.net

<http://packetlife.net/blog/2008/oct/16/pim-crash-course/>

<http://packetlife.net/blog/2010/jul/27/source-specific-multicast-pim-ssm/>

The Multicast Security Tool Kit

http://www.cisco.com/web/about/security/intelligence/multicast_toolkit.html

Configuring a Rendezvous Point

http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/rps.html

Configuring Multicast Source Discovery Protocol

http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfmsdp.html

Multicast Quick-Start Configuration Guide

<http://www.cisco.com/c/en/us/support/docs/ip/ip-multicast/9356-48.html>

Configuring IP Multicast Routing

http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfmulti.html

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_1_se/configuration/guide/3750xcg/swmcast.html

Good Luck

CCSI: Yasser Auda

<https://learningnetwork.cisco.com/people/yasser.r.a>

<https://www.facebook.com/YasserRamzyAuda>

<https://www.youtube.com/user/yasserramzyauda>